



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA

Máster en Ingeniería Informática

Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama

Alumno: Andrés Escobar Cotán



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA

Máster en Ingeniería Informática

Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama

Alumno: Andrés Escobar Cotán

Tutores: Helena Castán
Lanaspa
Salvador Dueñas Carazo

Agradecimientos

En este TFM, he recibido la ayuda y he tenido la confianza de numerosas personas a las que quiero expresar mi más sincero agradecimiento:

En primer lugar, agradecer a mis tutores Helena Castán y Salvador Dueñas por escogermme para este TFM, pero en especial por su continua confianza depositada, y animarnos durante la realización de este proyecto.

A mi amigo y compañero Eduardo Rodríguez, por todas las horas que hemos pasado en común desarrollando nuestros TFM.

A los profesores de la facultad que nos han ayudado con alguno de los puntos de este trabajo, en concreto a, Héctor García, Jesús Arias, y a Manuel Ángel González.

A mis compañeros y amigos del Máster, muy en especial a Verónica, por su apoyo durante la elaboración de esta memoria y por darme ánimos cuando más lo necesitaba. Aunque me gustaría agradecer también a David, a Miguel y a Amalia por los buenos momentos que como amigos hemos compartido.

Un agradecimiento muy especial a mis padres Eduardo y a Ana, así como a mi hermana Ana. Gracias por vuestra paciencia durante mi etapa universitaria.

A mis amigos Alejandro, Ricardo y Borja, gracias por animarme un poco la vida.

También agradecer a todos aquellos familiares y amigos que ya no están, y que habéis formado parte de este proceso.

Resumen

El objetivo de este Trabajo Fin de Máster, que lleva por título “Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama”, tiene como fin el estudio de dos instrumentos de alta importancia dentro de la electrónica como son el osciloscopio y el amplificador lock-in. Durante el desarrollo de este trabajo se expondrán los fundamentos técnicos de cada uno, para que sirven y sobre todo la existente relación entre ellos.

Para facilitar la lectura, un amplificador lock-in, es un dispositivo que permite a través de una señal alterna caracterizada por el ruido, extraer mediante la amplificación de la misma, las características de otra señal pura incrustada en la señal anterior y que se encuentra entre el ruido. Las características que permite obtener son la amplitud, la componente fase y la cuadratura. Un dispositivo de este tipo, es de gran utilidad en numerosos campos, como son el análisis de circuitos, o la caracterización de errores a la hora de fabricar materiales.

El osciloscopio es un dispositivo de medición, que permite caracterizar y visualizar diferentes tipos de onda.

La relación entre sendos dispositivos, el lock-in y el osciloscopio, será de utilidad de cara a presentar el software desarrollado, puesto que se pretenderá no solo desarrollar una solución que permita gestionar un osciloscopio, sino también implementar un lock-in software, que abarate los costes de adquisición de este tipo de instrumentos por parte de un laboratorio electrónico. Para ello el osciloscopio utilizado deberá presentar la posibilidad de generación de ondas, tal y como se verá más adelante, para que la generación sirva de entrada de referencia para el lock-in software.

El proyecto software presentado, además, tiene una característica de portabilidad, permitiendo añadir dispositivos de otras marcas, distintas a los dispositivos usados durante el desarrollo de este trabajo. Además, debido al lenguaje elegido C#, es posible el uso del proyecto en dispositivos embebidos suprimiendo así la necesidad de usar un ordenador de sobremesa.

Por último, citar que se ha implementado un modo de simulación que permite ejecutar experimentos teóricos, que facilita la comprensión de los dispositivos utilizados.

Finalmente, este proyecto es paralelo al presentado por mi compañero Eduardo Rodríguez Gutiez, centrándose el en los aspectos relativos al lock-in, y este en concreto en los relativos al osciloscopio y al generador de ondas.

Abstract

The aim of this dissertation, that has the following title “Arbitrary signal generation and current measure by a high-end digital oscilloscope”, has the purpose the study of two instruments of high importance in the electronics such like the oscilloscope and the lock-in amplifier. During the development of this work, the technical fundamentals of both instruments will be exposed, which are the common uses for both devices and over all the existing relationship between them.

For making easier the reading of this work, a lock-in amplifier, is a device that allows through an alternate signal characterised by the noise, extract via amplification of the signal itself, the characteristics of a pure signal that is inside in the previous signal and that is hidden by the noise.

In the set of characteristics that a lock-in amplifier could extract, there are the amplitude, the phase component and the quadrature one. A device of this kind, is very useful on large fields, such like are circuit analysis, or error characterization during material fabrication.

The oscilloscope, on the other hand, is a measurement device, that allows to characterise and visualize different kind of wavelength.

Relationship between both devices, the lock-in and the oscilloscope, will be of interest in order to present the software developed, because the aim of this project is not only for developing a software solution that allows to manage an oscilloscope, but also to implement a software lock-in, that enables to reduce the acquisition cost of this kind of instruments by an electronic laboratory. For that goal, the chosen oscilloscope must have the chance of generate a wavelength, so as is later presented in this work, the wave generated by the oscilloscope, could serve as an input reference for the lock-in implemented.

Also, the software project presented, has the attribute of portability, allowing to add different kinds of devices from another branches, rather than only the used ones during the development of this work. Further, due to the programming language chosen, C#, is possible the use of this software in embedded devices, thus suppressing the need of using a personal computer.

By last, mention that a simulation mode has been implemented, enabling the execution of theorists experiments, facilitating the understanding of the used devices.

Finally, this project was made parallel with the one presented by my mate Eduardo Rodríguez Gutiez, the focus of his work is related with the technical aspects of the lock-in device, while mine, is related to the oscilloscope and the wavelength generator ones.

Índice

- 1. Capítulo primero: INTRODUCCIÓN 1
 - 1.1. Descripción general de la aplicación 1
 - 1.1.1. Objetivos de la aplicación..... 1
 - 1.2. Contexto y necesidad de la aplicación..... 2
 - 1.3. Breve introducción a la tecnología usada 2
 - 1.4. Conclusión 2
 - 1.5. Organización de la memoria 3
- 2. Capítulo segundo: FUNDAMENTOS TÉCNICOS 5
 - 2.1 Principios básicos de señales y ondas 5
 - Suma de dos ondas senoidales..... 6
 - Representación Fasorial 7
 - Componente en fase y cuadratura..... 8
 - 2.2 El osciloscopio..... 8
 - Tipos de osciloscopio..... 9
 - Osciloscopios analógicos 9
 - Funcionamiento 10
 - Osciloscopios digitales con almacenamiento 10
 - Osciloscopios de señal mixta (MSO) 11
 - 2.2.1 Descripción del osciloscopio / generador Agilent DSO-X 3104A de Keysight. 12
 - Integridad de la señal 12
 - Panel frontal 12
 - Controles verticales 12
 - Controles horizontales 13
 - Controles del trigger o disparador 13
 - Controles de entrada 13
 - Teclas programables 14
 - Medidas básicas..... 14
 - Funciones matemáticas 15
 - Propiedades de rendimiento 15
 - Sondas 16
 - Propósito del osciloscopio / generador Agilent DSO-3104A 17
 - 2.2.2 Uso de los osciloscopios. 17
 - 2.3 El amplificador lock-in 17
 - 2.3.1 Principios de funcionamiento 17

Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama

Detección en banda estrecha.....	18
Señal de referencia del lock-in	18
Fase y magnitud	18
2.3.2 Descripción del amplificador lock-in SR844.....	19
Amplificador y filtros de entrada de señal	20
Canal de Referencia	21
Oscilador interno.....	21
Los detectores sensibles a la fase (PSD)	22
Constantes de tiempo y ganancia en corriente continua	23
Constantes de tiempos elevadas.....	24
Ganancia de salida	24
Reserva dinámica	24
Resolución	26
Salidas en corriente continua y escalado.....	27
Offset y expand	28
Ruido	28
Fuentes de ruido externas	29
Mediciones de ruido.....	32
2.3.3 Usos del amplificador Lock-in.....	33
Detectores de gas	33
Espectroscopia de fluorescencia	34
Espectroscopia de impedancia electroquímica	34
Resonancia paramagnética electrónica.....	34
Resonancia magnética nuclear.....	35
Balanzas.....	36
Microscopía de fuerza atómica	36
Medición del sonido	38
3. Capítulo tercero: GESTIÓN DEL PROYECTO	39
3.1 Definiciones	39
3.2. Perspectiva y objetivo del proyecto	41
3.3. Participantes o Stakeholders del proyecto	41
3.4. Costes del proyecto	41
Costes Software	42
Costes de personal.....	43
3.5. Planificación del proyecto	44
3.5.1. Iteraciones en el proyecto.....	44

Primera iteración (1 de octubre – 20 de octubre)	44
Segunda iteración (21 de octubre- 17 de noviembre)	44
Tercera iteración (18 de noviembre – 10 de diciembre de 2015).....	44
Cuarta iteración (10 de diciembre de 2015 – abril 2016)	45
3.5.2. Riesgos	45
Categorías de riesgos	45
Tabla de riesgos	46
3.5.2.1. Monitorización de riesgos	48
3.5.3. Calendario del proyecto	49
4. Capítulo cuarto: ANÁLISIS DE LA APLICACIÓN	53
4.1. Glosario	53
4.2. Actores	54
4.3. Análisis de requisitos	54
Requisitos Funcionales	54
Requisitos No Funcionales.....	59
4.4. Casos de Uso	61
UC-001: Crear Onda.....	62
UC-002: Modificar Onda	64
UC-003: Borrar Onda	66
UC-004: Cargar Onda.....	68
UC-005: Guardar Onda.....	70
UC-006: Extraer Características.....	72
UC-007: Conectar.....	74
UC-008: Desconectar.....	76
UC-009: Actualizar Datos.....	78
UC-010: Cambiar Canal	80
UC-011: Enviar Onda	82
4.5. Modelo de Dominio	84
4.6. Diagrama de clases de análisis	84
5. Capítulo quinto: DISEÑO DE LA APLICACIÓN	86
5.1. Objetivos del diseño	86
5.2. Arquitectura Lógica.....	87
Elección de la arquitectura.....	87
Diagrama de la arquitectura lógica del sistema.....	88
5.3. Arquitectura física del sistema	89
5.4. Descripción de los casos de uso de diseño	91

Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama

UC-001: Crear Onda	91
UC-002: Modificar Onda	94
UC-003: Borrar Onda.....	97
UC-004: Cargar Onda	100
UC-005: Guardar Onda	103
UC-006: Extraer Características	105
UC-007: Conectar	107
UC-008: Desconectar	109
UC-009: Actualizar Datos	111
UC-010: Cambiar Canal	114
UC-011: Enviar Onda	116
5.5. Diagrama de clases de diseño	118
5.6. Diseño de la interfaz	119
Diseños iniciales	119
Diseño Final.....	123
6. Capítulo sexto: IMPLEMENTACIÓN DE LA APLICACIÓN	126
6.1. Primera Iteración	126
6.2. Segunda Iteración	129
6.3. Tercera Iteración.....	132
6.3.1. Proyecto ‘ProyectoRefactorizado’	132
6.3.1.1. Enviar Onda	133
Cálculo de la suma de ondas.....	133
Código de envío	135
6.3.1.2. Actualizar Datos	136
6.3.1.3. Reconstrucción de ondas.....	139
Filtrado de ruido.....	139
Reconstrucción de la amplitud.....	141
Reconstrucción de la frecuencia	141
Reconstrucción del desfase.....	141
6.4. Cuarta Iteración	144
Almacenar y recuperar ondas en formato XML	144
Proyecto Esqueleto Inicial.....	146
7. Capítulo séptimo: Pruebas y resultados	150
7.1. Pruebas de caja negra.....	150
UC-001 – Crear Onda	150
UC-002 – Modificar Onda.....	152

Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama

UC-003 – Borrar Onda.....	152
UC-004 – Cargar Onda	152
UC-005 – Guardar Onda	153
UC-006 – Extraer Características	154
UC-007 – Conectar	154
UC-008 – Desconectar.....	155
UC-009 – Actualizar Datos.....	155
UC-0010 – Cambiar Canal.....	156
7.2. Pruebas de generación y recepción de ondas	157
Montaje del experimento genérico	157
Montaje para las pruebas de generación y recepción de ondas	158
Prueba 1.....	159
Prueba 2.....	160
Prueba 3.....	162
8. Capítulo Octavo: Conclusiones	164
8.1. Dificultades encontradas.....	164
8.2. Objetivos alcanzados	165
8.3. Líneas de trabajo futuro	166
8.4. Conclusiones	167
Contenido del CD-ROM.....	168
Bibliografía	169

Índice de figuras

ILUSTRACIÓN 1-REPRESENTACIÓN GRÁFICA DE UNA ONDA SINUSOIDAL	6
ILUSTRACIÓN 2-SUMA DE DOS ONDAS: (0.5V Y 800 HZ, Y 1V Y 1GHZ).....	7
ILUSTRACIÓN 3-REPRESENTACIÓN FASORIAL Y EUCLEDIANA DE LAS ONDAS Y1, Y2 Y SU SUMA Y	8
ILUSTRACIÓN 4-FOTOGRAFÍA DE UN OSCILOSCOPIO ANALÓGICO	9
ILUSTRACIÓN 5-REPRESENTACIÓN DEL ESQUEMA DE FUNCIONAMIENTO DE UN OSCILOSCOPIO	10
ILUSTRACIÓN 6-DIAGRAMA DE BLOQUES, DEL FUNCIONAMIENTO DE UN OSCILOSCOPIO DIGITAL	11
ILUSTRACIÓN 7- FOTOGRAFÍA DEL PANEL FRONTAL DEL AGILENT DSO-X 3104A ...	12
ILUSTRACIÓN 8-ESQUEMA DE BLOQUES DE UN LOCK-IN.....	19
ILUSTRACIÓN 9- REPRESENTACIÓN DE RELACIÓN ENTRE LA FRECUENCIA Y LA RESERVA DEL LOCK-IN	25
ILUSTRACIÓN 10-PANEL FRONTAL Y TRASERO DE UN AMPLIFICADOR LOCK-IN.....	27
ILUSTRACIÓN 11-ESQUEMA DEL EXPERIMENTO DE ACOPLAMIENTO CAPACITIVO, QUE INCLUYE AL RUIDO	30
ILUSTRACIÓN 12-ESQUEMA DEL EXPERIMENTO DE ACOPLAMIENTO INDUCTIVO...	31
ILUSTRACIÓN 13-ESQUEMA DEL EXPERIMENTO DE ACOPLAMIENTO RESISTIVO	31
ILUSTRACIÓN 14- ESPECTRÓMETRO EPR EN LA UNIVERSIDAD JAGUELÓNICA, POLONIA. FOTOGRAFÍA DE PRZEMYSŁAW “TUKAN”	35
ILUSTRACIÓN 15-MONTAJE DE UN ESPECTRÓMETRO DE RESONANCIA MAGNETICA NUCLEAR (NMR) EN EL LABORATORIO DE EXPERIMENTACIÓN	35
ILUSTRACIÓN 16- DIAGRAMA DE BLOQUES DEL SENSOR CAPACITIVO DE MASA	36
ILUSTRACIÓN 17- DIAGRAMA DE FUNCIONAMIENTO DE UN MICROSCOPIO DE FUERZA ATÓMICA.....	37
ILUSTRACIÓN 18- FOTOGRAFÍA CON EL MONTAJE PARA MEDIR LA VELOCIDAD DEL SONIDO USANDO LOCK-IN	38
ILUSTRACIÓN 19 - DIAGRAMA DE GANTT DEL CALENDARIO DEL PROYECTO	52
ILUSTRACIÓN 20- DIAGRAMA DE CASOS DE USO	61
ILUSTRACIÓN 21- DIAGRAMA DE SECUENCIA DE ANÁLISIS DEL CASO DE USO 001 ..	63
ILUSTRACIÓN 22-DIAGRAMA DE SECUENCIA DE ANÁLISIS DEL CASO DE USO 002 ..	66
ILUSTRACIÓN 23-DIAGRAMA DE SECUENCIA DE ANÁLISIS DEL CASO DE USO 003 ..	67
ILUSTRACIÓN 24-DIAGRAMA DE SECUENCIA DE ANÁLISIS DEL CASO DE USO 004 ..	70
ILUSTRACIÓN 25-DIAGRAMA DE SECUENCIA DE ANÁLISIS DEL CASO DE USO 005 ..	72
ILUSTRACIÓN 26- DIAGRAMA DE SECUENCIA DE ANÁLISIS DEL CASO DE USO 006 ..	73
ILUSTRACIÓN 27- DIAGRAMA DE SECUENCIA DE ANÁLISIS DEL CASO DE USO 007 ..	76
ILUSTRACIÓN 28- DIAGRAMA DE SECUENCIA DE ANÁLISIS DEL CASO DE USO 008 ..	77
ILUSTRACIÓN 29- DIAGRAMA DE SECUENCIA DE ANÁLISIS DEL CASO DE USO 009 ..	80
ILUSTRACIÓN 30-DIAGRAMA DE SECUENCIA DE ANÁLISIS DEL CASO DE USO 010 ..	81
ILUSTRACIÓN 31- DIAGRAMA DE SECUENCIA DE ANÁLISIS DEL CASO DE USO 011 ..	84
ILUSTRACIÓN 32- DIAGRAMA DEL MODELO DE DOMINIO	84
ILUSTRACIÓN 33- DIAGRAMA DE CLASES DE ANÁLISIS	85
ILUSTRACIÓN 34- DIAGRAMA DE LA ARQUITECTURA LÓGICA.....	88
ILUSTRACIÓN 35- DIAGRAMA DE LA ARQUITECTURA FÍSICA	90
ILUSTRACIÓN 36- DIAGRAMA DE SECUENCIA DE DISEÑO DEL CASO DE USO 001	92
ILUSTRACIÓN 37- DIAGRAMA DE SECUENCIA DE DISEÑO DEL CASO DE USO 001 (PARTE DOS)	93
ILUSTRACIÓN 38- DIAGRAMA DE SECUENCIA DE DISEÑO DEL CASO DE USO 002	95

ILUSTRACIÓN 39- DIAGRAMA DE SECUENCIA DE DISEÑO DEL CASO DE USO 002 (PARTE DOS).....	96
ILUSTRACIÓN 40- DIAGRAMA DE SECUENCIA DE DISEÑO DEL CASO DE USO 003	98
ILUSTRACIÓN 41- DIAGRAMA DE SECUENCIA DE DISEÑO DEL CASO DE USO 003 (PARTE DOS).....	99
ILUSTRACIÓN 42- DIAGRAMA DE SECUENCIA DE DISEÑO DEL CASO DE USO 004 ..	101
ILUSTRACIÓN 43- DIAGRAMA DE SECUENCIA DE DISEÑO DEL CASO DE USO 004 (PARTE DOS).....	102
ILUSTRACIÓN 44- DIAGRAMA DE SECUENCIA DE DISEÑO DEL CASO DE USO 005 ..	104
ILUSTRACIÓN 45-DIAGRAMA DE SECUENCIA DE DISEÑO DEL CASO DE USO 006...	106
ILUSTRACIÓN 46-DIAGRAMA DE SECUENCIA DE DISEÑO DEL CASO DE USO 007 ...	108
ILUSTRACIÓN 47-DIAGRAMA DE SECUENCIA DE DISEÑO DEL CASO DE USO 008...	110
ILUSTRACIÓN 48- DIAGRAMA DE SECUENCIA DE DISEÑO DEL CASO DE USO 009 ..	112
ILUSTRACIÓN 49- DIAGRAMA DE SECUENCIA DE DISEÑO DEL CASO DE USO 009 (PARTE DOS).....	113
ILUSTRACIÓN 50- DIAGRAMA DE SECUENCIA DE DISEÑO DEL CASO DE USO 010 ..	115
ILUSTRACIÓN 51- DIAGRAMA DE SECUENCIA DE DISEÑO DEL CASO DE USO 011 ..	117
ILUSTRACIÓN 52- DIAGRAMA DE CLASES DE DISEÑO	118
ILUSTRACIÓN 53 - MOCK-UP DONDE SE MUESTRA LA SELECCIÓN DE INSTRUMENTOS	119
ILUSTRACIÓN 54 - MOCK-UPS DONDE SE MUESTRA EL MENÚ DE SIMULACIÓN	120
ILUSTRACIÓN 55 – MOCK-UP DE LA VISTA DEL OSCILOSCOPIO GENÉRICO	120
ILUSTRACIÓN 56-MOCK-UP DONDE SE MUESTRA LA INTERFAZ DEL GENERADOR DE SEÑAL GENÉRICO	121
ILUSTRACIÓN 57- MOCK-UP DEL GENERADOR DE ONDAS	121
ILUSTRACIÓN 58 - MOCK-UP DEL AMPLIFICADOR LOCK-IN	122
ILUSTRACIÓN 59 - MOCK-UP DEL REGISTRADOR DE DATOS	122
ILUSTRACIÓN 60- INTERFAZ DE LA APLICACIÓN, NADA MÁS SER EJECUTADA.....	123
ILUSTRACIÓN 61- VENTANA PRINCIPAL DE LA APLICACIÓN CON UNA ONDA RECIBIDA DESDE EL CANAL 2	124
ILUSTRACIÓN 62- DESCRIPCIÓN DE AYUDA PARA EL BOTÓN “ACTUALIZAR”	124
ILUSTRACIÓN 63- INTERFAZ FINAL DEL MEZCLADOR DE ONDAS.....	125
ILUSTRACIÓN 64 - ASPECTO DE LA INTERFAZ DE LA SEGUNDA ITERACIÓN	130
ILUSTRACIÓN 65- DIAGRAMA DE CLASES DE LA SEGUNDA ITERACIÓN	131
ILUSTRACIÓN 66- CAPTURA DE LA SEÑAL SIN ZOOM	139
ILUSTRACIÓN 67-AMPLIACIÓN DE LA SEÑAL ANTERIOR, HACIA LA MITAD, CON UNAS DECENAS DE MUESTRAS	139
ILUSTRACIÓN 68-DEMOSTRACIÓN DEL ALGORITMO DEL “FARO”	140
ILUSTRACIÓN 69- DIAGRAMA DE RELACIÓN ENTRE EL NÚMERO PI Y ÁNGULOS CORRESPONDIENTES	143
ILUSTRACIÓN 70 - ESQUEMA DE BLOQUES DEL PROYECTO: "ESQUELETO INICIAL"	146
ILUSTRACIÓN 71- ESQUEMA DE CLASES CORRESPONDIENTE A LA GESTIÓN DE ONDAS.....	147
ILUSTRACIÓN 72- ASPECTO DE LA INTERFAZ DEL PROYECTO "ESQUELETOINICIAL"	148
ILUSTRACIÓN 73- DIAGRAMA DE CLASES PROPUESTO PARA LA CUARTA ITERACIÓN	149
ILUSTRACIÓN 74- FOTOGRAFÍA QUE ILUSTRA EL EXPERIMENTO GENÉRICO DEL OSCILOSCOPIO, CORTESÍA DE FRANCISCO JAVIER PEÑA RUBIO	157
ILUSTRACIÓN 75- FOTOGRAFÍA QUE ILUSTRA EL EXPERIMENTO PARA LA GENERACIÓN T RECEPCIÓN DE ONDAS	158

ILUSTRACIÓN 76 - FOTOGRAFÍA DE LA ONDA GENERADA POR LA APLICACIÓN
PARA LA PRUEBA 1 159

ILUSTRACIÓN 77-FOTOGRAFÍA DE LA ONDA GENERADA POR EL OSCILOSCOPIO
PARA LA PRUEBA 1 160

ILUSTRACIÓN 78-FOTOGRAFÍA DE LA ONDA GENERADA POR LA APLICACIÓN PARA
LA PRUEBA 2 161

ILUSTRACIÓN 79-FOTOGRAFÍA DE LA ONDA GENERADA POR EL OSCILOSCOPIO
PARA LA PRUEBA 2 161

ILUSTRACIÓN 80-FOTOGRAFÍA DE LA ONDA GENERADA POR LA APLICACIÓN PARA
LA PRUEBA 3 162

ILUSTRACIÓN 81- FOTOGRAFÍA DE LA ONDA GENERADA POR EL OSCILOSCOPIO
PARA LA PRUEBA 3 163

Índice de ecuaciones

ECUACIÓN 1- ONDA SINUSOIDAL 6

ECUACIÓN 2- SUMA DE DOS ONDAS SINOUIDALES SIN DESFASE 7

ECUACIÓN 3-SUMA DE DOS ONDAS SINOUIDALES CON DESFASE 7

ECUACIÓN 4- SUMA DE DOS ONDAS SINOUIDALES CON DISTINTA FRECUENCIA Y
AMPLITUD 7

ECUACIÓN 5- REPRESENTACIÓN FASORIAL DE UNA SINUSOIDE 8

ECUACIÓN 6- ECUACIÓN DE SALIDA DEL PSD 17

ECUACIÓN 7- SALIDA DE UN LOCK-IN BIFÁSICO 18

ECUACIÓN 8- SALIDA DEL LOCK-IN CONTANDO CON LA OPERACIÓN DE EXPAND 28

ECUACIÓN 9- RUIDO TÉRMICO 29

ECUACIÓN 10- RUIDO DE DISPARO 29

ECUACIÓN 11- CORRIENTE PARÁSITA 30

ECUACIÓN 12- EFECTOS MICROFÓNICOS 32

Índice de tablas

TABLA 1- COSTES HARDWARE	42
TABLA 2- COSTES SOFTWARE.....	43
TABLA 3- COSTES DE PERSONAL.....	43
TABLA 4 - RIESGO 001	46
TABLA 5- RIESGO 002	46
TABLA 6 - RIESGO 003	47
TABLA 7- RIESGO 004	47
TABLA 8 - RIESGO 005	47
TABLA 9 - RIESGO 006	48
TABLA 10 - RIESGO 007	48
TABLA 11 - CALENDARIO DEL PROYECTO	50
TABLA 12- ACTOR USUARIO.....	54
TABLA 13- ACTOR OSCILOSCOPIO	54
TABLA 14 - REQUISITO FUNCIONAL 001.....	54
TABLA 15 - REQUISITO FUNCIONAL 002.....	55
TABLA 16 - REQUISITO FUNCIONAL 003.....	55
TABLA 17 - REQUISITO FUNCIONAL 004.....	55
TABLA 18 - REQUISITO FUNCIONAL 005.....	55
TABLA 19 -REQUISITO FUNCIONAL 006.....	56
TABLA 20 - REQUISITO FUNCIONAL 007.....	56
TABLA 21- REQUISITO FUNCIONAL 008.....	56
TABLA 22- - REQUISITO FUNCIONAL 009	57
TABLA 23- REQUISITO FUNCIONAL 010.....	57
TABLA 24- REQUISITO FUNCIONAL 011.....	57
TABLA 25- - REQUISITO FUNCIONAL 012	57
TABLA 26- REQUISITO FUNCIONAL 013.....	58
TABLA 27- REQUISITO FUNCIONAL 014.....	58
TABLA 28-- REQUISITO FUNCIONAL 015	58
TABLA 29-- REQUISITO FUNCIONAL 016	58
TABLA 30- REQUISITO NO FUNCIONAL 001	59
TABLA 31- REQUISITO NO FUNCIONAL 002	59
TABLA 32- REQUISITO NO FUNCIONAL 003	59
TABLA 33- REQUISITO NO FUNCIONAL 004	59
TABLA 34- REQUISITO NO FUNCIONAL 005	60
TABLA 35- DESCRIPCIÓN DE ANÁLISIS DEL CASO DE USO 001	62
TABLA 36- DESCRIPCIÓN DE ANÁLISIS DEL CASO DE USO 002	64
TABLA 37- DESCRIPCIÓN DE ANÁLISIS DEL CASO DE USO 003	66
TABLA 38- DESCRIPCIÓN DE ANÁLISIS DEL CASO DE USO 004	68
TABLA 39- DESCRIPCIÓN DE ANÁLISIS DEL CASO DE USO 005	70
TABLA 40- DESCRIPCIÓN DE ANÁLISIS DEL CASO DE USO 006	72
TABLA 41- DESCRIPCIÓN DE ANÁLISIS DEL CASO DE USO 007	74
TABLA 42- DESCRIPCIÓN DE ANÁLISIS DEL CASO DE USO 008	76
TABLA 43- DESCRIPCIÓN DE ANÁLISIS DEL CASO DE USO 009	78
TABLA 44- DESCRIPCIÓN DE ANÁLISIS DEL CASO DE USO 010	80
TABLA 45- DESCRIPCIÓN DE ANÁLISIS DEL CASO DE USO 011	82
TABLA 46 - DESCRIPCIÓN DE DISEÑO DEL CASO DE USO 001	91
TABLA 47- DESCRIPCIÓN DE DISEÑO DEL CASO DE USO 002	94
TABLA 48- DESCRIPCIÓN DE DISEÑO DEL CASO DE USO 003	97

TABLA 49- DESCRIPCIÓN DE DISEÑO DEL CASO DE USO 004.....	100
TABLA 50- DESCRIPCIÓN DE DISEÑO DEL CASO DE USO 005.....	103
TABLA 51- DESCRIPCIÓN DE DISEÑO DEL CASO DE USO 006.....	105
TABLA 52- DESCRIPCIÓN DE DISEÑO DEL CASO DE USO 007.....	107
TABLA 53- DESCRIPCIÓN DE DISEÑO DEL CASO DE USO 008.....	109
TABLA 54- DESCRIPCIÓN DE DISEÑO DEL CASO DE USO 009.....	111
TABLA 55- DESCRIPCIÓN DE DISEÑO DEL CASO DE USO 010.....	114
TABLA 56- DESCRIPCIÓN DE DISEÑO DEL CASO DE USO 011.....	116
TABLA 57- CÓDIGO DE LA PRIMERA ITERACIÓN	128
TABLA 58- CÓDIGO CORRESPONDIENTE A SUMAR DOS ONDAS.....	134
TABLA 59- CÓDIGO CORRESPONDIENTE AL ENVÍO DE DATOS AL OSCILOSCOPIO ..	135
TABLA 60 - CÓDIGO CORRESPONDIENTE A LA ACTUALIZACIÓN DE DATOS EN LAS CLASES DEL MODELO	137
TABLA 61- CÓDIGO CORRESPONDIENTE A LA ACTUALIZACIÓN DE DATOS EN LAS CLASES DE LA VISTA.....	138
TABLA 62- CÓDIGO CORRESPONDIENTE A LA ACTUALIZACIÓN DE DATOS USANDO UN DELEGADO	138
TABLA 63- CÓDIGO PARCIAL DE EXTRAER CARACTERÍSTICAS	141
TABLA 64- CÓDIGO PARCIAL DE EXTRAER CARACTERÍSTICAS Y RENDERIZAR	142
TABLA 65- CÓDIGO COMPLETO DE EXTRAER CARACTERÍSTICAS.....	144
TABLA 66- EJEMPLO DEL ESQUEMA XML PARA ALMACENAMIENTO DE ONDAS....	145
TABLA 67- DESCRIPCIÓN DE LAS PRUEBAS A REALIZAR.....	150
TABLA 68- PRUEBA DEL CASO DE USO 001	151
TABLA 69- PRUEBA DEL CASO DE USO 002	152
TABLA 70- PRUEBA DEL CASO DE USO 003	152
TABLA 71- PRUEBA DEL CASO DE USO 004	153
TABLA 72- PRUEBA DEL CASO DE USO 005	154
TABLA 73- PRUEBA DEL CASO DE USO 006	154
TABLA 74- PRUEBA DEL CASO DE USO 007	155
TABLA 75- PRUEBA DEL CASO DE USO 008	155
TABLA 76- PRUEBA DEL CASO DE USO 009	156
TABLA 77- PRUEBA DEL CASO DE USO 010	157
TABLA 78- DESCRIPCIÓN DE LA PRUEBA DE GENERACIÓN Y RECEPCIÓN DE ONDAS	159
TABLA 79- PRIMERA PRUEBA	159
TABLA 80- SEGUNDA PRUEBA.....	160
TABLA 81-TERCERA PRUEBA.....	162

1. Capítulo primero: INTRODUCCIÓN

El proyecto presentado a continuación surge de parte de la Universidad de Valladolid, en concreto de nuestros tutores Helena Castán y Salvador Lanaspá con el siguiente objetivo; desarrollar una aplicación que permita generar señales arbitrarias y obtener las características de la corriente a través de un osciloscopio tras haberla pasado previamente por distintos circuitos básicos. El resultado de este experimento es de gran interés para la aplicación científica tal y como puede ser en campos como la creación de placas solares, el estudio del consumo eléctrico, o simplemente la enseñanza.

Cabe citar que este proyecto en sí es una de las dos partes que componen el proyecto original, siendo esta la primera, que consiste en la implementación de un sistema software que permitiera la generación de señales arbitrarias y su posterior recogida y análisis a través de un osciloscopio. La segunda parte del proyecto, centrada en la implementación de un dispositivo Lock-In en software fue presentada por mi compañero Eduardo Rodríguez Gutiez [1]

1.1.Descripción general de la aplicación

La siguiente aplicación está orientada a la ayuda de control de dispositivos de medición para departamentos de electrónica, en concreto, de dispositivos de tipo osciloscopio y generadores de señal. Los dispositivos seleccionados han de contar con una interfaz compatible USB, para comunicarse con la aplicación, en concreto y dado la carencia de estándares en este aspecto, la aplicación está preparada para dispositivos de la marca Keysight, [2] aunque gracias a la potencia del paradigma de la orientación al objeto, es fácil su extensión a otros dispositivos de otras marcas usando para ello un patrón adaptador [3].

La finalidad última, explicada con más exactitud en el trabajo final de Máster de Eduardo Rodríguez Gutiez es la de combinar sendos instrumentos para realizar un Lock-In virtual [1].

En último lugar, es necesario citar que no es necesario utilizar un dispositivo en sí para utilizar la aplicación, puesto que se ha desarrollado un simulador tanto del generador como del osciloscopio que permite la toma y el estudio de datos.

1.1.1. Objetivos de la aplicación

Los objetivos del sistema son los siguientes:

- Desarrollo de un simulador de osciloscopio.
- Desarrollo de un simulador de generador de señal.
- Desarrollo de un sistema autónomo de dispositivo de control de osciloscopio.
- Desarrollo de un sistema autónomo de dispositivo de control de generadores de señal.
- Combinar ambos sistemas para proveer de un Lock-In virtual.

1.2. Contexto y necesidad de la aplicación

Actualmente el grupo de investigación: “Grupo de Caracterización de Materiales y Dispositivos Electrónicos [4]”, posee diferente instrumentación electrónica que realiza la tarea de medición y facilita la interpretación de los datos de los diferentes estudios que realiza el grupo. Sin embargo, esta instrumentación es cara, y se necesita proveer por tanto de herramientas virtuales que permitan reducir costos y facilitar la movilidad y portabilidad de los mismos.

En concreto, aunque existen osciloscopios virtuales elaborados en LabView [5] disponibles en el mercado, se carece de un amplificador “Lock-In” virtual, siendo este, un tipo de amplificador que permite la extracción de una señal de origen en un entorno ruidoso.

Para el desarrollo de este “lock-in” virtual, se va a utilizar la combinación de osciloscopio y generador de señal citado anteriormente.

1.3. Breve introducción a la tecnología usada

Para el desarrollo de este proyecto, se ha optado por un lenguaje orientado a objetos, en concreto C# bajo .NET Framework, aunque en electrónica se suele utilizar lenguajes visuales como LabView, se eligió este esquema, por la capacidad de la orientación al objeto de extensión e independencia del hardware y por haber trabajado previamente con estas tecnologías a lo largo de la carrera. Se profundizará más en este apartado en el capítulo tercero de esta memoria.

En resumen, las tecnologías usadas han sido:

- Windows 7 y superiores
- Visual Studio 2013
- .Net Framework 4.0
- WPF (Windows Presentation Foundation)
- Agilent Keysight Drivers [6]

1.4. Conclusión

El amplificador lock-in es de gran utilidad en el campo de la electrónica en la investigación de células solares y en otros campos, desafortunadamente el coste de este instrumento es elevado por lo que se precisa de un sistema software que permita simular las tareas del mismo. El objetivo de este proyecto es por tanto el de como a partir de un osciloscopio y un generador de señal, apoyados por software se es capaz de crear un amplificador lock-in.

A mayores y para garantizar la independencia de dispositivo, se ha trabajado en simular no solo el lock-in sino también el osciloscopio y el generador de señal. Otro apartado que garantiza la independencia es que en el mercado los instrumentos virtuales existentes no hacen uso de la orientación del objeto, que proporciona un mecanismo de abstracción facilitando que la aplicación desarrollada se pueda utilizar de manera sencilla simplemente instalando los drivers apropiados.

1.5. Organización de la memoria

Los contenidos de la presente memoria se encuentran distribuidos de acuerdo al siguiente esquema:

- **Capítulo 1:** Introducción
- **Capítulo 2:** Fundamentos teóricos:
 - Donde se explica los teoremas matemáticos relacionados con la generación de señales arbitrarias, medidas de corrientes de bajo nivel, así como el uso del osciloscopio Agilent DSO-X 3104A [7] de la compañía Keysight.

Los capítulos del tercero al sexto hacen referencia a apartados relacionados con la calidad del software, así como de la ingeniería del software del proyecto que se presenta. Comenzando con la metodología usada, continuando con la planificación para empezar con el análisis y diseño de la aplicación y acabar finalmente con la implementación de la misma.

- **Capítulo 3:** Gestión del proyecto
 - Este capítulo aborda la planificación y la metodología de calidad del software seguida en el proyecto. A su vez también trata aspectos económicos como el cálculo de costes. Se compone de los siguientes subapartados:
 - Metodología usada.
 - Participantes del proyecto.
 - Costes
 - Hardware.
 - Software.
 - Equipo humano.
 - Planificación.
 - Gestión de riesgos.
 - Herramientas utilizadas.
- **Capítulo 4:** Análisis de la aplicación
 - Aborda la fase de análisis desde el punto de vista del modelo Waterfall, incorporando además de la gestión de requisitos, los casos de uso y diagramas de secuencia característicos del modelo Open UP. El capítulo tiene los siguientes sub apartados:
 - Definición de requisitos.
 - Definición de casos de uso.
 - Diagrama de secuencia.
- **Capítulo 5:** Diseño de la aplicación
 - Aborda la fase de diseño del modelo Open UP, definiendo las diferentes arquitecturas y las clases que lo componen, por último, la parte enfocada a la interacción se trata también en este capítulo. Se compone de los siguientes sub apartados:
 - Arquitectura lógica y física del sistema.

- Descripción de los casos de uso.
- Diseño de clases
- Diseño de la interfaz de usuario.
- **Capítulo 6:** Implementación de la aplicación
 - En último lugar se desarrollan los aspectos más interesantes desde el punto de vista técnico de la implementación, tales como problemas encontrados con las herramientas utilizadas, así como las soluciones empleadas para resolver las ecuaciones planteadas en este tipo de sistemas. El capítulo tiene los siguientes sub apartados:
 - Implementación de la interfaz
 - Generador de onda arbitraria
 - Medida de corrientes de bajo nivel
 - Comunicación con el osciloscopio
- **Capítulo 7:** Pruebas y resultados.
 - El capítulo séptimo concluye la parte destinada de esta memoria a los principios establecidos en la Ingeniería del Software, con pruebas de caja negra y caja blanca realizados a la aplicación, así como una serie de resultados obtenidos y su interpretación en el contexto de la electrónica.
- **Capítulo 8:** Conclusiones.

En último lugar en el séptimo capítulo, se establece una serie de conclusiones y líneas de trabajo para futuras investigaciones. Se compone de los siguientes sub-apartados:

 - Dificultades encontradas.
 - Conocimientos adquiridos.
 - Trabajo futuro.
- **Contenido del CD-ROM**
- **Bibliografía**

2. Capítulo segundo: FUNDAMENTOS TÉCNICOS

El propósito de este capítulo es el de dar a conocer los fundamentos técnicos necesarios para comprender el proyecto.

En este capítulo se van a exponer los siguientes apartados:

- Principios básicos de señales y ondas.
- El osciloscopio.
 - Definición.
 - Principio de funcionamiento
 - Descripción del osciloscopio / generador Agilent DSO-X 3104A de Keysight.
- El lock-in.
 - Definición.
 - Principio de funcionamiento
 - Descripción del amplificador lock-in [8].

2.1 Principios básicos de señales y ondas

En este apartado se introducen una serie de conocimientos técnicos básicos [9] que permiten la comprensión total de este capítulo.

Señal: Una señal se define como la variación de una magnitud física con respecto a otra. En función del tipo de magnitud física que varia se puede distinguir entre señal lumínica, eléctrica, sonora...

Sensor: Un sensor es aquel dispositivo que permite mediante la distinción de dichas magnitudes capturar esas señales. A mayores son capaces de traducir estas señales a variables eléctricas, facilitando su interpretación y transmisión.

Filtro pasa-baja: Es un filtro eléctrico que atenúa las frecuencias más altas a una dada y permite las frecuencias más bajas que esta.

Onda: Una **onda** es un fenómeno vibratorio consistente en la propagación de una perturbación de alguna propiedad o magnitud física a través de un medio.

Existen **ondas periódicas**, es decir, que presentan el mismo comportamiento a lo largo del tiempo. Y por el contrario también existen ondas no periódicas.

Las ondas sinusoidales son las ondas periódicas más comunes, se les llama sinusoidal porque representa gráficamente a la función seno.

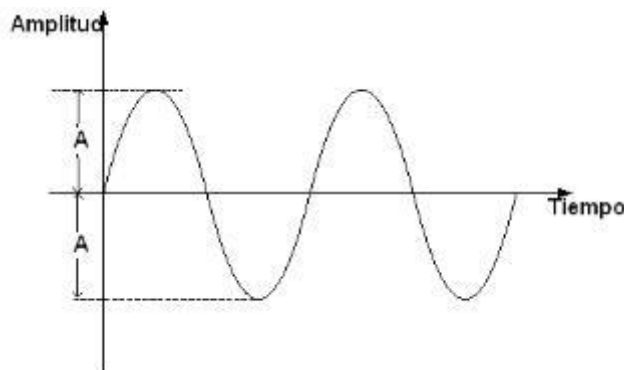


Ilustración 1-Representación gráfica de una onda sinusoidal, tomado de [10]

Estas ondas, tienen las siguientes características:

- **Amplitud:** La diferencia entre el valor más alto que alcanza, también conocido como pico, respecto a su valor medio.
- **Valle:** La distancia entre el valor más bajo que alcanza respecto a su valor medio.
- **Amplitud pico a pico:** La distancia entre existente entre los valores de un valle y un pico. Si el valle presenta el mismo valor absoluto que el pico, entonces la amplitud pico a pico se corresponde con el doble de la amplitud.
- **Periodo:** Es el tiempo en que la onda completa una oscilación.
- **Frecuencia:** Es el inverso del periodo, e indica el número de oscilaciones que la onda realiza por segundo.

Se puede describir una onda sinusoidal matemáticamente como [11]:

$$y(x) = A \sin\left(\frac{2\pi}{T}x + \varphi\right)$$

Ecuación 1- Onda sinusoidal

Donde:

- A es la amplitud de oscilación.
- T es el período.
- φ es la fase inicial.

Se dice que una onda es **compuesta**, cuando se transmite más de una señal por un canal, produciéndose la suma de las ondas simples que componen dicha señal. De estas ondas simples, habrá una onda que presente la misma frecuencia que la compuesta original, a la frecuencia que presenta esta onda se le etiquetará como **frecuencia fundamental** mientras que a las restantes anteriores se le llamará **armónicos**.

Suma de dos ondas senoidales

Si dos ondas senoidales de la misma frecuencia se encuentran en fase, siendo cada una de ellas:

$$y_1 = A_1 \sin(wt) \text{ y } y_2 = A \sin(wt)$$

Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama

Con $w = \frac{2\pi}{T}$

Entonces **su suma será:**

$$y = y_1 + y_2 = (A_1 + A_2) \text{sen}(wt)$$

Ecuación 2- Suma de dos ondas sinusoidales sin desfase

Si las ondas por el contrario están **desfasadas**, una respecto a la otra, con un ángulo de desfase ϕ . Sus ecuaciones serán:

$$y_1 = A_1 \text{sen}(wt) \text{ y } y_2 = A_2 \text{sen}(wt - \phi)$$

Ecuación 3-Suma de dos ondas sinusoidales con desfase

Y su suma será [11]:

$$y = y_1 + y_2 = (A_1 + A_2) \text{sen}(wt)$$

Por último, para sumar dos ondas sinusoidales con **distinta frecuencia y distinta amplitud**, se aplica la siguiente fórmula matemática:

$$y = y_1 + y_2 = (A_1 - A_2) \cos(\omega t - kx) + 2A_2 \cos(\omega t) \cos(kx)$$

Ecuación 4- Suma de dos ondas sinusoidales con distinta frecuencia y amplitud

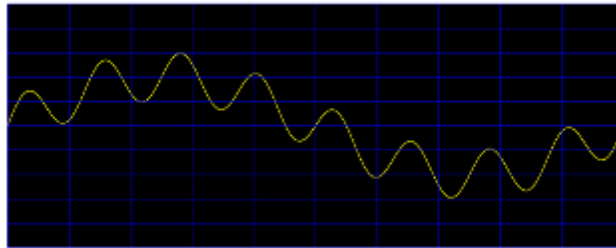


Ilustración 2-Suma de dos ondas: (0.5V y 800 Hz, y 1V y 1GHz)

Representación Fasorial

Fasor: Se define a fasor como la **representación gráfica** de un número complejo. Este tipo de representación se utiliza en ingeniería electrónica y en telecomunicaciones para representar señales.

Si una onda sinusoidal tiene la siguiente ecuación matemática:

$$y = A \text{sen}(wt + \theta)$$

Siendo θ el ángulo de fase.

Esta ecuación también puede expresarse de la siguiente manera:

$$y = I(Y) = I(A(\cos(wt + \theta) + i \text{sen}(wt + \theta)))$$

Donde i es la unidad imaginaria definida como $i = \sqrt{-1}$.

E, $I(Y)$ es la parte imaginaria del número complejo Y .

Utilizando la fórmula de Euler se puede re-escribir el fasor de la siguiente manera:

$$y = A e^{i(wt + \theta)}$$

Por tanto, la representación fasor de esta senoide sería:

$$Y = Ae^{i\theta}$$

Ecuación 5- Representación fasorial de una senoide

Por tanto, el fasor Y es un número complejo constante que contiene la magnitud y fase de la senoide.

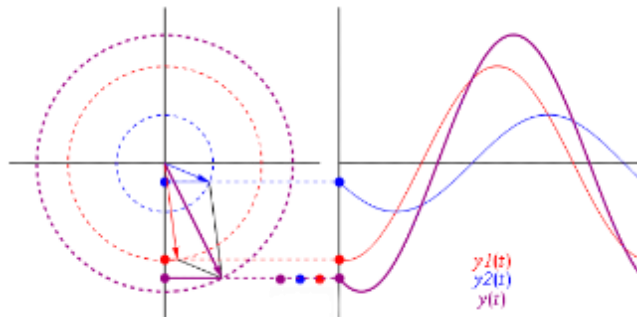


Ilustración 3-Representación fasorial y Euclidiana de las ondas y_1 , y_2 y su suma y [11]

Componente en fase y cuadratura

De cara a interpretar una onda, resulta de gran interés definir el concepto de fase.

Una fase, se define como el desplazamiento horizontal respecto a un punto de referencia que representa la onda senoide en un instante t de tiempo. Geométricamente si se representa el ciclo como una circunferencia de 360° , la fase representa el ángulo actual que tiene la onda en un instante t dado.

Cuadratura se define como el estado relativo en el que se encuentran dos ondas sinusoidales cuya diferencia de constantes es igual a 90° . Re-escribiéndolo en función del termino fase, se dice que la cuadratura se presenta cuando una onda tiene fase 1 y la otra tiene fase 0.

2.2 El osciloscopio

Un osciloscopio [12] se define como un instrumento que mide variaciones de una señal eléctrica a lo largo del tiempo y que traduce dicha variación a un formato visual que representa en una pantalla.

Para facilitar la interpretación de los datos, el osciloscopio traduce lo detectado a un espacio geométrico, donde por normal general el tiempo representa el eje horizontal y el voltaje el eje vertical.

Inicialmente los osciloscopios eran analógicos, disminuyendo su fiabilidad, estos se basaban en un tubo de rayos catódicos (CRT). Estos tubos poseen un cañón de electrones que excitado eléctricamente desprende electrones por termo emisión desde su cátodo hasta su ánodo. Cuando una señal necesita ser visualizada, pasa por un amplificador interno del osciloscopio que produce una deflexión vertical y horizontal de las distintas bobinas que controlan el haz de electrones, dibujando así una onda en la pantalla.

Los actuales osciloscopios modernos son digitales, y utilizan un procesado digital de señales, poseen por tanto de un conversor analógico digital (DAC) que impreso en un circuito transforma el

Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama

valor continuo del voltaje del canal en un valor discreto binario que posteriormente procesa un microcontrolador.

Tipos de osciloscopio

Existen en la actualidad tres tipos de osciloscopios [13] analógicos, de almacenamiento digital (DSO) y de señal mixta (MSO).

Osciloscopios analógicos



Ilustración 4-Fotografía de un osciloscopio analógico

Los primeros osciloscopios utilizan tubos de rayos catódicos para mostrar una forma de onda. El fósforo luminiscente de la pantalla se ilumina cuando un electrón le golpea, y tras sucesivos bits del fósforo se encienden se puede ver una representación de la señal. Un trigger (disparador) es necesario para que la onda mostrada sea estable.

Cuando una traza entera de la pantalla es completada, el osciloscopio espera hasta que un evento específico ocurra y luego comienza la traza de nuevo.

Una pantalla sin usar un trigger es inestable puesto que la forma de onda no es mostrada como estable.

Los osciloscopios basados en pantallas de fosforo tienen una utilidad asociada, que es, debido al fosforo la onda mostrada no desaparece inmediatamente, sino que permite que dibujos en la pantalla sucesivos se solapen mostrando fallos o irregularidades en la señal. También la intensidad con la que un electrón golpea la pantalla está correlacionada con la intensidad de la señal actual. Esto hace que la pantalla actúe como un lienzo tridimensional (en otras palabras, el eje X se correspondería al tiempo, el Y al voltaje y la Z a la intensidad).

Como desventaja asociada, el osciloscopio analógico no puede congelar la imagen y mantener la forma de onda por un periodo de tiempo extenso. Una vez que la sustancia de fosforo empieza a perder su capacidad de iluminar, una parte de la señal se pierde.

Otra desventaja es que no se pueden hacer mediciones automáticas, en su lugar hay que utilizar el mallado de la pantalla. Además, muchos tipos de señales no pueden ser mostradas por que la frecuencia de barrido horizontal y vertical del dibujado es lenta.

Funcionamiento

Los osciloscopios [12] analógicos como se ha citado anteriormente utilizan un tubo de rayo catódicos para su funcionamiento.

Para ello al tubo están conectadas unas placas que mueven al rayo en los ejes Y e X, estas son las placas de desviación vertical y horizontal, tal y como se muestra en la siguiente figura:

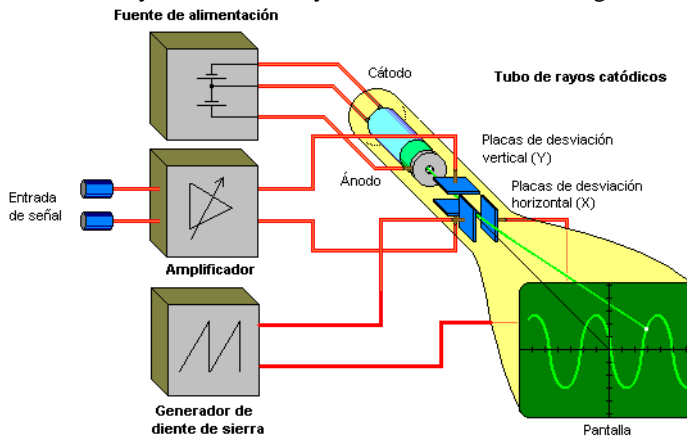


Ilustración 5-Representación del esquema de funcionamiento de un osciloscopio [12]

Las placas horizontales, que se encargan de mover a lo largo del eje X el rayo catódico, generan una tensión con forma de diente de sierra a través de un circuito oscilador apropiado cuya frecuencia es regulable, permitiendo así adaptarse a la frecuencia de la señal a medir. En cambio, la placa vertical se conecta a un amplificador con ganancia ajustable y alta impedancia de entrada.

El tubo de rayos catódicos el rayo de electrones creado por en el cátodo y acelerado por el ánodo choca en la pantalla, la cual está recubierta por una capa fluorescente que se ilumina por el impacto de los electrones.

Cuando se aplica una diferencia de potencial en cualquiera de las dos placas de desviación, se desvía el haz de electrones generado en el cátodo, debido al campo eléctrico que ha creado la diferencia de potencial. Al contrario, cuando no se aplica ninguna señal, el haz del cátodo se moverá en dirección horizontal (gracias al generador de diente de sierra y la placa horizontal) dibujando una línea recta, al llegar al extremo físico de la pantalla, volverá al punto inicial para realizar un nuevo barrido. Gracias a la velocidad con la que se realiza esta última parte y al apagado parcial del rayo, el ojo humano no puede percibir el retorno.

Al estar los ejes de coordenadas marcados en la pantalla, es posible establecer, como se ha citado anteriormente, relaciones entre las divisiones, siendo el eje X el periodo y el eje Y el voltaje. Cada división en cada uno de los ejes se corresponderá por tanto a un valor de tiempo o voltaje concreto.

Osciloscopios digitales con almacenamiento

Los osciloscopios digitales con almacenamiento [13, p. 9] (conocidos también como DSOs) fueron inventados para remediar muchos de los negativos aspectos de los osciloscopios analógicos.

Los DSOs reciben una señal y la digitalizan utilizando un conversor analógico-digital conocido como DAC. La siguiente figura muestra la arquitectura utilizada por los osciloscopios de la marca Agilent-Keysight:

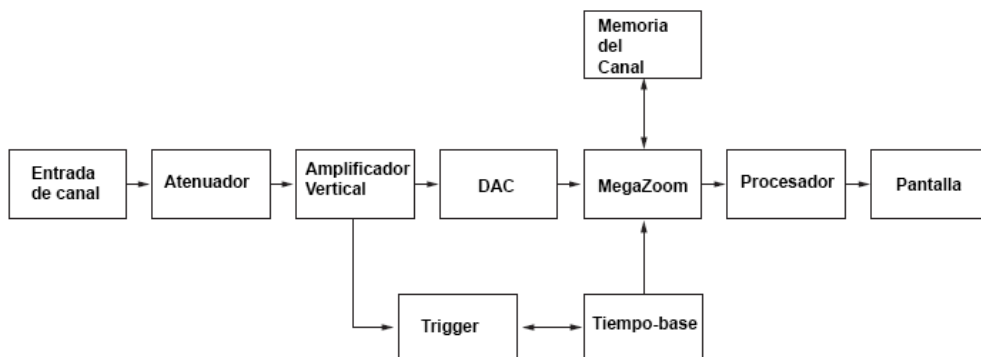


Ilustración 6-Diagrama de bloques, del funcionamiento de un osciloscopio digital, extraído de [13, p. 9]

Los bloques que lo componen son los siguientes:

Entrada de canal: Lugar donde se conecta la sonda, hacia la señal a medir.

Atenuador: Escala o atenúa la señal que se está midiendo.

Amplificador Vertical: Proporciona escalado adicional a nivel de voltaje, antes de pasar la señal al DAC.

DAC: Conversor analógico-digital, toma muestras de la señal y las digitaliza, una vez realizado almacena los datos en la memoria.

Trigger: Comprueba y acciona los disparadores del osciloscopio.

Tiempo-base: Ajusta el tiempo de la pantalla para el osciloscopio.

MegaZoom: Característica comercial de los osciloscopios de Agilent-Keysight que permite mostrar una señal por tenue que sea de manera clara y escalarla automáticamente en la pantalla.

Procesador: Proporciona determinado filtrado de la señal en pos procesamiento antes de mostrarlo por pantalla.

Tener los datos en formato digital habilita la posibilidad de realizar medidas mayores en la onda. Además, pueden ser guardadas permanentemente en la memoria, y los datos pueden ser imprimidos o transferidos a un ordenador o a un dispositivo de almacenamiento extraíble.

Osciloscopios de señal mixta (MSO)

Normalmente en un DSO, la señal de entrada es analógica y el DAC la digitaliza. Sin embargo, a día de hoy es necesario medir no solo señales analógicas de un circuito sino también digitales (ej: transferencia de datos en un bus), por lo que es necesario dispositivos que permitan medir tanto señales digitales como analógicas. Este dispositivo es lo que se conoce como MSO [13, p. 9], a diferencia de los DSO proporcionan cuatro canales analógicos y un número mayor de canales digitales. Además, se posee la ventaja de ser capaces de proporcionar combinaciones de ambos tipos de señal y mostrarles todas juntas, correlacionados en un mismo tiempo base.

2.2.1 Descripción del osciloscopio / generador Agilent DSO-X 3104A de Keysight.

Integridad de la señal

El propósito principal de un osciloscopio es el de proporcionar una representación visual precisa de las señales eléctricas. Por esta razón el concepto de integridad de la señal es muy importante.

La integridad de la señal se define como la habilidad del osciloscopio para reconstruir la señal de onda de tal manera que es una representación fidedigna de la señal original. Un osciloscopio con baja integridad es inútil porque carece de sentido realizar un test cuando la forma de onda del osciloscopio no posee la forma o características de la señal verdadera. Es, sin embargo, importante recordar que un osciloscopio nunca va a representar una forma de onda de manera exacta, esto es debido a que cuando se conecta un osciloscopio a un circuito, el osciloscopio se convierte parte del circuito, introduciendo efectos de carga. Aunque los fabricantes intentan minimizar estos efectos siempre existen en algún grado.

Panel frontal

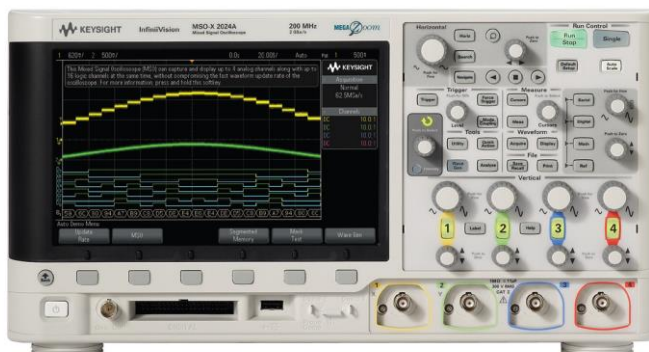


Ilustración 7- Fotografía del panel frontal del Agilent DSO-X 3104A

El panel frontal del osciloscopio Agilent [14, pp. 36-40] [7] presenta las siguientes secciones:

- Las entradas de canal
- La pantalla
- Los controles horizontales
- Los controles verticales
- Las teclas programables (Softkeys)
- Y los controles de los trigger o disparadores

Las señales se envían al osciloscopio mediante las entradas de canal, que son en realidad conectores para las sondas del experimento. La pantalla es simplemente el lugar donde las señales son mostradas, en el caso del osciloscopio que nos ocupa es una pantalla LCD en color.

Controles verticales

Los controles verticales de un osciloscopio [14, pp. 51-67] [15] están agrupados en la sección denominada vertical. Estos controles permiten ajustar aspectos verticales de la pantalla. Entre ellos

Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama

asignar el número de voltios por división del eje Y, permitiendo realizar así un zoom en la forma de onda aumentando o disminuyendo esta cantidad. También hay un control sobre el offset, esto es, realizar un desplazamiento en el eje Y de la forma de onda.

Controles horizontales

En un osciloscopio, [13] los controles horizontales están agrupados en la sección marcada como horizontal. Estos permiten realizar ajustes en la escala horizontal de la pantalla. En concreto y análogo al caso vertical, permiten ajustar la división de tiempos en el eje X. Y al igual que en el vertical permite realizar un zoom en un rango en el tiempo. También se permite ajustar el retardo horizontal, es decir, escanear a través de un intervalo de tiempo determinado.

Controles del trigger o disparador

Como se ha mencionado anteriormente en esta memoria, disparar en una señal permite proporcionar una salida estable y usable para mediciones. Esto permite sincronizar la adquisición de la parte de la onda que es interesante para medir. Los controles del disparador permiten establecer el nivel de trigger vertical (el voltaje a partir del cual se quiere disparar) y escoger entre varias capacidades de triggering, en concreto [13, p. 14]:

- **Edge triggering**
 - El trigger se dispara cuando el voltaje sobrepasa algún valor de umbral. En el DSO-3014A se puede escoger entre un “edge” ascendente o otro descendente.
- **Glitch triggering**
 - Este modo habilita la posibilidad de disparar un evento o pulso cuyo ancho es mayor o menor que un tiempo específico. Es de gran utilidad para encontrar fallos en la onda, si estos fallos no ocurren demasiado, será difícil verlos.
- **Pulse-width triggering**
 - Es similar al anterior, la diferencia radica en que se pueden disparar en pulsos de cualquier ancho y polaridad (negativa o positiva), además se puede establecer la posición horizontal del trigger. Esto permite ver que ha ocurrido antes del trigger o a posteriori. Por instancia, se puede ejecutar el disparador de “glitch”, encontrar el error, y entonces usar este trigger para ver que ha causado el fallo. Si se tiene el retraso horizontal en cero, el evento de disparado será pintado en el medio horizontal de la pantalla. Eventos que ocurran justo antes del disparador aparecerán a la izquierda de la pantalla y los de después aparecerán a la izquierda. También se puede establecer el acoplamiento del disparador y la fuente de entrada a la que disparar, en último lugar el DSO-3104A permite utilizar este trigger en una señal relacionada.

Controles de entrada

Hay típicamente entre dos y cuatro canales analógicos en un osciloscopio, en el caso del DSO-3104A son cuatro. Estos están enumerados y tienen un botón asociado que permite su activación y desactivación, además hay una selección que permite seleccionar entre acoplamiento en continua o en corriente alterna.

Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama

Si el acoplamiento en corriente continua es seleccionado, la señal entera será mostrada, en caso contrario (acoplamiento en corriente alterna) bloqueará la componente en continua y centrará la forma de onda en 0 voltios. De manera adicional se puede especificar la impedancia de la sonda para cada canal a través de un botón de selección. También los controles permiten elegir entre dos maneras de muestreo:

- **Muestreo en tiempo real:**
 - El muestreo en tiempo real simplifica la forma de onda de tal manera que captura una completa imagen de la forma de onda en cada adquisición. Los osciloscopios actuales permiten capturar hasta formas de onda de 32 GHz en un único disparo en tiempo real. El DSO-3104A permite hasta 20 GHz.
- **Muestreo en tiempo equivalente:**
 - En este modo el osciloscopio va dibujando la forma de onda a través de numerosas adquisiciones. Se muestrea una parte de la onda en la primera adquisición, otra en la segunda, etcétera. Entonces une toda esta información para recrear la forma de onda. El muestreo en tiempo equivalente es útil para señales de alta frecuencia que son muy rápidas para el muestreo en tiempo real (mayores de 32 GHz).

Teclas programables

Las teclas programables son encontradas en osciloscopios que no llevan un sistema operativo de Windows embebido. Estas teclas permiten navegar a través del sistema de menús mostrado en la pantalla del osciloscopio, para navegar a través de las opciones es necesario pulsar estas teclas de manera continua.

Medidas básicas

Los osciloscopios digitales permiten realizar un amplio rango de medidas sobre la forma de onda. La complejidad y rango de las medidas dependen de las características disponibles en el osciloscopio.

Las medidas básicas más comunes son [13, p. 14]:

- **Voltaje pico a pico:** Esta medida calcula el voltaje entre el voltaje más bajo y el más alto de un ciclo de la forma de onda.
- **Voltaje RMS:** Esta medida calcula el voltaje RMS de la forma de onda.
- **Tiempo de alzado (Risetime):** Esta medida calcula el tiempo que lleva a una señal desde su voltaje más bajo, a un voltaje alto. Esto se calcula normalmente computando el tiempo que lleva a la onda desde el 10% al 90% de voltaje pico a pico.
- **Ancho de pulso:** Esta medida computa el ancho de un pulso calculando el tiempo que lleva a la onda de ir desde un 50% del valor pico a pico al máximo voltaje y volver al 50%.
- **Periodo:** Calcula el periodo de la forma de onda.
- **Frecuencia:** Esta medida calcula la frecuencia de la forma de onda.

Funciones matemáticas

En adición a las medidas, los osciloscopios digitales permiten calcular automáticamente operaciones matemáticas sobre las ondas. Algunas de ellas son:

- **Suma o resta:** Permite sumar o restar múltiples formas de onda y mostrar la señal resultante.
- **Integración:** Computa la integral de la forma de onda.
- **Transformada de Fourier:** Esta función permite ver las frecuencias que componen la señal.
- **Valor absoluto:** Esta función muestra el valor absoluto (en términos de voltaje) de la forma de onda.

Propiedades de rendimiento

Los osciloscopios digitales poseen propiedades que pueden afectar al rendimiento del instrumento y a su capacidad de medida. Algunas de estas propiedades son:

- **Ancho de banda:** El ancho de banda es la característica más importante de un osciloscopio, da al usuario una indicación de su rango de frecuencias con el que el osciloscopio trabaja. En otras palabras, determina el rango de señales que se pueden mostrar y medir de manera precisa. El ancho de banda se mide en hertzios, y se mide como la frecuencia más baja a la que la señal de entrada es atenuada por 3 dB.
- **Canales:** Un canal se refiere a una entrada independiente en el osciloscopio.
- **Tasa de muestreo:** La tasa de muestreo de un osciloscopio es el número de muestras que puede adquirir un osciloscopio por segundo. Es recomendable que el osciloscopio tenga una tasa mínima de 2.5 veces mayor que el ancho de banda. El fabricante normalmente especifica la tasa máxima que puede alcanzar el osciloscopio, pero esa tasa solamente es posible cuando uno o dos canales están siendo usados. Si más canales están siendo usados simultáneamente la tasa de muestreo descenderá. Es importante destacar que para dibujar la onda completa, el osciloscopio interpola los puntos obtenidos mediante muestreo utilizando: $\frac{\sin(x)}{x}$.
- **Capacidad de la memoria (Memory depth):** Como se ha mencionado anteriormente, un osciloscopio digital utiliza un conversor DAC para digitalizar la onda proveniente de un canal de entrada. Los datos digitalizados se almacenan en una memoria de alta velocidad del osciloscopio. La capacidad de la memoria se refiere a cuantos puntos y en consecuencia la longitud de tiempo que estos pueden ser almacenados. La capacidad de la memoria, aunque fija se ve empujada por el resto de parámetros, en concreto por el tiempo de muestreo, si se solicita muestrear a una onda durante varios segundos, el número de puntos que se almacenen será menor que si el tiempo de captura fuera más pequeño. Matemáticamente [14] se puede formular esta relación de la siguiente manera:

$$\text{Capacidad de la memoria} = (\text{Tasa de muestreo}) * (\text{División de tiempo})$$

- **Tasa de actualización:** La tasa de actualización se refiere a la tasa a la cual un osciloscopio puede adquirir y actualizar la pantalla de una forma de onda. Existe un tiempo muerto entre adquisiciones de la onda, durante este tiempo muerto una porción de la onda es no mostrada en el osciloscopio. Como resultado, si algún evento infrecuente o fallo ocurre durante alguno de esos tiempos, no se podrá ver. Tener una tasa de actualización rápida implica, por tanto, tener menores tiempos muertos.

Sondas

Para poder conectar el osciloscopio al experimento que queremos medir, es necesario el uso de sondas. Las sondas son cruciales para obtener una señal medible, para ello es preciso conocer las siguientes características:

- **Carga:** Ninguna sonda es capaz de reproducir perfectamente la señal, esto es debido a que la sonda a la hora de medir se convierte en parte de ese circuito. Parte de la energía eléctrica pasa a través de la sonda. Este fenómeno es la carga, hay tres tipos de carga:
 - **Resistiva:** Puede causar que la amplitud de la señal sea incorrecta. Puede incluso provocar que un circuito deje de funcionar. Es bueno comprobar la resistencia interna de la sonda, en concreto lo ideal es que sea 10 veces superior la resistencia interna de la sonda respecto a la que ofrece el circuito, para obtener una reducción de la amplitud menor al 10%.
 - **Capacitiva:** Provoca que los tiempos de alzado se ralenticen y el ancho de banda se reduzca. Para reducir la carga capacitiva, la sonda ha de tener 5 veces el ancho de banda de la señal.
 - **Carga Inductiva:** Aparecen provocando una disminución y desfase entre las señales. Esto ocurre por efectos inductivos en el cable de la tierra de la sonda, por lo que es mejor utilizar el cable más corto posible.

A mayores se puede realizar la siguiente clasificación entre las sondas [13, pp. 22-23]:

- **Sondas Pasivas:** Las sondas pasivas contienen solo componentes pasivas y no requieren alimentación externa para su operación. Son útiles para medir señales con anchos de bandas 600 MHz. Son de coste reducido, fáciles de usar y versátiles. Estas sondas producen alta capacidad de carga y baja resistividad de carga.
- **Sondas Activas:** Estas sondas requieren alimentación externa, normalmente utilizan una conexión USB. Las sondas tienen componentes electrónicos activos (es decir aquellos que excitan el circuito, diodos, transistores...) para amplificar o condicionar una señal. Son capaces de soportar anchos de banda mucho más altos. Se distinguen de las sondas pasivas en que son más caros y pesadas. Sin embargo, ofrecen una mejor combinación de carga resistiva y capacitiva.

- **Sondas de corriente:** Estas ondas son utilizadas para medir el flujo de la corriente en el circuito. Tienen un ancho de banda relativamente pequeño (100 MHz)

Propósito del osciloscopio / generador Agilent DSO-3104A

Un osciloscopio [15] como se ha citado anteriormente, es un instrumento de medida y prueba de una variable concreta en función de otra. También el osciloscopio Agilent puede realizar otras tareas más allá de dibujar el voltaje sobre el tiempo. Los canales permiten mezclar ambos voltajes o comparar uno con otro usando el modo XY del osciloscopio. Esto es útil cuando se dibujan graficas I-V o patrones de Lissajous para detectar la fase y la ratio de frecuencia entre sendas señales.

2.2.2 Uso de los osciloscopios.

- Determinar directamente el periodo y el voltaje de una señal.
- Determinar indirectamente la frecuencia de una señal.
- Determinar que parte de la señal es DC y cual AC.
- Localizar averías en un circuito.
- Medir la fase entre dos señales.
- Determinar que parte de la señal es ruido y como varia este en el tiempo

2.3 El amplificador lock-in

Un lock-in [16] [17] es un dispositivo electrónico que permite detectar y medir pequeñas señales de corriente alterna, con una precisión del orden de nano-voltios. Incluso en condiciones poco favorables en que la señal a medir, sea ocultada por numerosas fuentes de ruido con mayor amplitud que la señal, el lock-in es capaz frente a otros dispositivos de obtener dicha señal. El principio fundamental que caracteriza a este dispositivo es que utiliza una técnica, conocida como detección de fase para extraer la componente de la señal requerida a una específica frecuencia y fase.

2.3.1 Principios de funcionamiento

Los lock-in necesitan una frecuencia de referencia para funcionar [18]. Por lo general cuando se realiza un experimento con el lockin, se excita a este último a una señal determinada con un generador de funciones. El lockin, a su vez genera una señal sinusoidal con la misma frecuencia y un desfase.

A posterior el lock-in amplifica la señal de entrada y la multiplica entonces por la referencia del lock-in utilizando un detector sensible a la fase o multiplicador.

La salida del detector de fase sería por tanto la multiplicación de dos ondas sinusoidales en AC:

$$V_{PSD} = \frac{1}{2} V_{SIG} V_L \cos([w_R - w_L]t + \theta_{SIG} - \theta_{REF}) - \frac{1}{2} V_{SIG} V_L \cos([w_R + w_L]t + \theta_{SIG} + \theta_{REF})$$

Si la salida se pasa a través de un filtro pasa baja, se logran eliminar las señales AC, si las señales tenían la misma frecuencia la salida quedará en una corriente DC proporcional con el siguiente valor:

$$V_{PSD} = \frac{1}{2} V_{SIG} V_L \cos(\theta_{SIG} - \theta_{REF})$$

Ecuación 6- Ecuación de salida del PSD

Detección en banda estrecha

Si la señal de entrada está afectada por el ruido, el PSD y el filtro de pasa baja solamente permitirán detectar señales cuyas frecuencias sean muy cercanas a la frecuencia del lock-in.

Las señales de ruido a frecuencias mayores que las de entrada quedarían mermadas a la salida del PSD por el filtro pasa baja. Esta atenuación depende del ancho de banda y del roll-off (es decir la pendiente que existe entre la banda de paso y la banda de rechazo) del filtro pasa baja.

Señal de referencia del lock-in

Es necesario que el lock-in presente una señal de referencia con la misma frecuencia que la señal del experimento. Además, esta señal de referencia ha de presentar la misma fase que la señal de entrada, y en caso de que esta última varíe a lo largo del tiempo, dicha variación ha de trasladarse a la señal de referencia del lock-in.

Por lo tanto, es necesario que ambas señales estén sincronizadas, para lograr este objetivo el lock-in utiliza un bucle de enganche de fase (PLL, phase-locked loop) para generar la señal de referencia.

El proceso es el siguiente, se aporta al lock-in una señal externa de referencia (a través de un generador de funciones, una onda cuadrada). El PLL une el oscilador de referencia interno (que genera señales sinusoidales frente a cuadradas) con la señal externa de referencia, oscilando a ω_{Ref} con un desfase fijo de θ_{Ref} , ya que el PLL vigila activamente la referencia externa, los cambios en la frecuencia de esta última no afectan a la medición.

Fase y magnitud

La salida del PSD es proporcional a $V_{SIG} \cos \theta$ donde $\theta = (\theta_{SIG} - \theta_{REF})$. Este valor sería la diferencia existente entre la fase de la señal de entrada y la del oscilador de referencia del lock-in.

Dado que θ_{REF} es un valor controlable, al ajustarse a cero se produce que $V_{SIG} \cos \theta = V_{SIG}$

Por lo que el valor del coseno sería 0, al contrario, si ajustamos el desfase de referencia para que este 90° respecto a θ_{SIG} la salida será nula.

Un lock-in [19] puede ser de dos tipos:

- **Monofásicos:** Si cuenta con un único PSD.
- **Bifásico:** Si cuenta con dos PSD, conectados en paralelo, de tal manera que cada uno de ellos representa una fase del procesamiento.

En caso de tener un lock-in bifásico, la dependencia de fase podría ser eliminada dando una salida aproximada con la siguiente ecuación:

$$V_{PSD2} = \frac{1}{2} V_{SIG} V_L \sin(\theta_{SIG} - \theta_{REF})$$

Ecuación 7- Salida de un lock-in bifásico

Por lo tanto, al final del proceso se tendrían dos salidas, una proporcional a $\cos \theta$ y otra a $\sin \theta$

Si se denomina, $X = \cos \theta$ e $Y = \sin \theta$, se podría decir que X representa a la componente en fase de la señal a recuperar e Y a la componente en cuadratura.

Con X e Y se pueden extraer los siguientes datos aplicando las siguientes ecuaciones:

$$\begin{aligned} \text{Amplitud:} \quad V_{SIG} &= \sqrt{X^2 + Y^2} \\ \text{Fase:} \quad \theta &= \tan^{-1}\left(\frac{Y}{X}\right) \end{aligned}$$

2.3.2 Descripción del amplificador lock-in SR844

La siguiente imagen, muestra el diagrama de bloques del amplificador lock-in DSP SR850, de la marca SRS. Aunque en esta memoria se hablará del modelo anterior el DSP SR844 [8] que es el que poseen en el laboratorio del grupo de Caracterización de Materiales y Dispositivos Eléctricos.

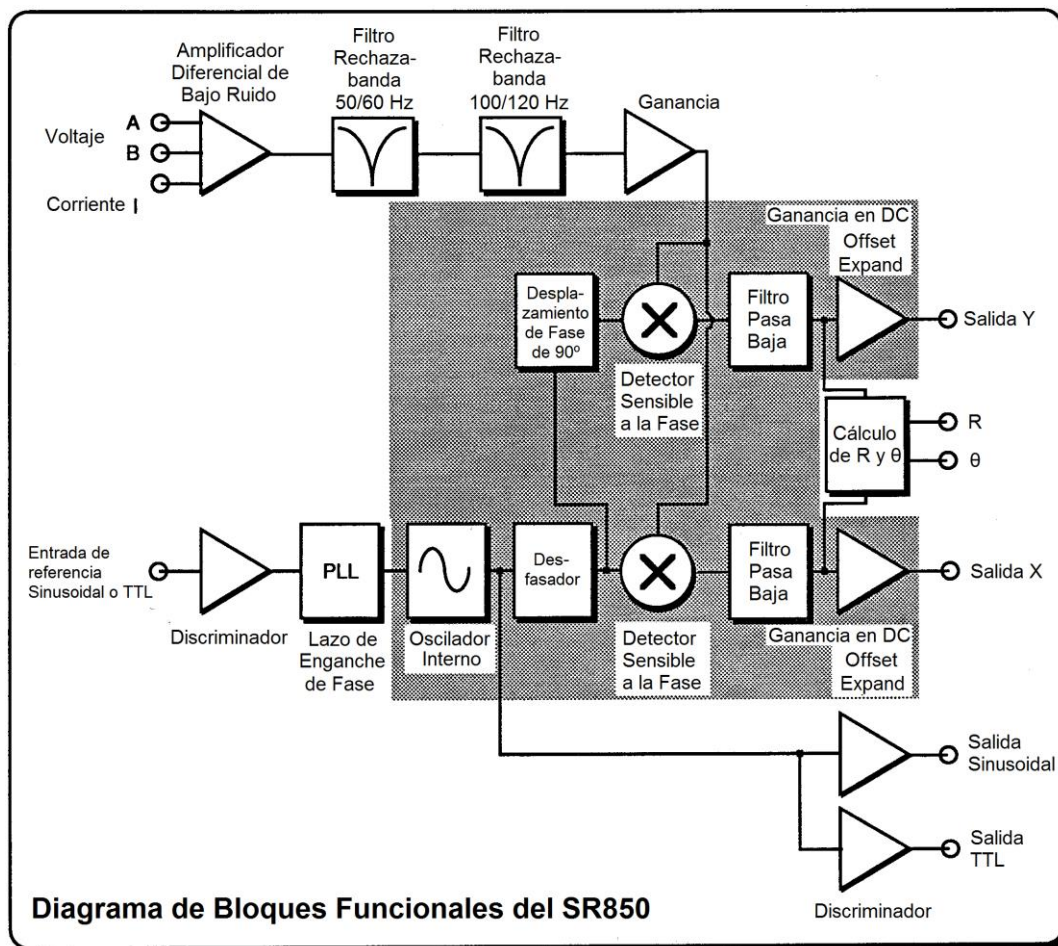
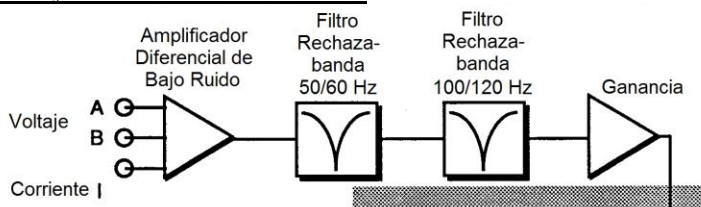


Ilustración 8-Eschema de bloques de un lock-in, extraído de [20, p. 59]

A continuación, se detallarán cada uno de estos bloques funcionales.

Amplificador y filtros de entrada de señal



Como ya se ha citado anteriormente, un lock-in puede medir señales tan pequeñas del orden de nanos voltios, para poder hacer esto es necesario un amplificador de señal de bajo ruido para elevar la señal a un nivel alto con el cual el lock-in pueda trabajar sin degradar la señal de ruido.

La ganancia analógica del lockin SR844 varía entre 7 y 1000, como se comentará a posterior, el control de la ganancia es fundamental puesto que ganancias mayores no mejorarán la relación señal ruido y no sería necesaria.

La ganancia global de la señal alterna más la señal continua está delimitada por un concepto conocido como sensibilidad. La distribución de la ganancia viene dada por la reserva dinámica.

Ambos conceptos se detallarán en profundidad en las páginas posteriores.

Filtros rechaza banda

El SR844 tiene dos filtros rechaza banda en la cadena de amplificación de señal. Están configurados para trabajar a la frecuencia de la red eléctrica (50 o 60 Hz) y a su doble. Cuando la señal de entrada presente señales fuentes de ruido a la frecuencia de la red eléctrica, estos filtros se pueden activar para eliminar dichas señales. Eliminar este ruido antes de la primera etapa de ganancia, garantizará como se verá más adelante mayor reserva dinámica para poder obtener una mejor medición.

Sin embargo, el usar estos filtros presentan una clara desventaja, y es que, si se quieren realizar mediciones en un rango de 10 Hz próximo a estas frecuencias, por ejemplo, si se realizan medidas a 70 Hz activar el filtro de 60 Hz atenuaría la señal y la medida será errónea.

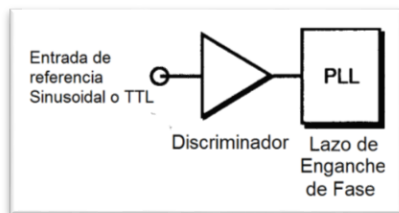
Filtro de suavizado.

Después de la etapa de ganancia y antes de entrar al PSD la señal de entrada pasa por un filtro de suavizado. Esto es necesario, para como se verá más adelante poder digitalizar la señal del lock-in. En función del criterio de Nyquist, estas señales deben ser muestreadas a una frecuencia que sea como mínimo el doble de la frecuencia más alta de la señal.

En el caso del lock-in SR844 la frecuencia de señal más alta es de 100 kHz y la de muestreo sería de 256 kHz, sin embargo, no se debe permitir que cualquier señal por encima de 128 kHz alcance el conversor, dado que violarían el criterio de Nyquist y serían submuestreadas. El resultado de este submuestreo es que estas señales de mayor frecuencia quedarían como si fueran de frecuencias inferiores en el flujo de datos digital.

Para poder corregir esto, es necesario filtrar la señal y eliminar cualquier señal por encima de 154 kHz, se añade por tanto otro filtro de suavizado que elimine las frecuencias comprendidas entre 102 kHz y 154 kHz logrando una atenuación por encima de los 154 kHz al menos de 100 dB. Este filtro es transparente al usuario por lo que no afectarían a la medición final.

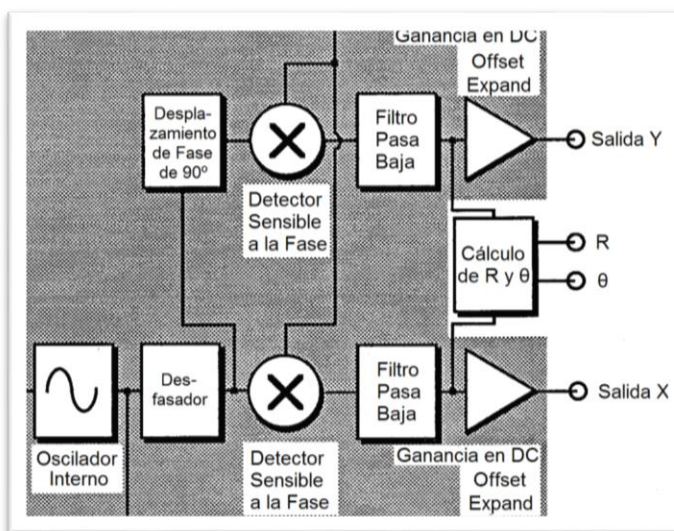
Canal de Referencia



En este canal, se introduce la señal de referencia generada por un oscilador, en nuestro caso el DSO-X 3104^a enganchado en fase a la frecuencia de la señal.

La entrada de este canal, tiene una impedancia de entrada de $1M\Omega$ y ha de presentar una frecuencia por encima de 1Hz.

Oscilador interno



El oscilador del lockin es un generador de funciones de 100 kHz con posibilidad de sincronización con onda cuadrada y salida de señal sinusoidal. Este oscilador puede engancharse a la fase externa de referencia.

Esta señal sinusoidal no es pura, sino que es una señal sintetizada de manera digital. El DSP (procesador digital de señales) se encarga de traducir los valores calculados del seno para enviarlos a un conversor digital analógico. Finalmente, un filtro de suavizado produce una señal sinusoidal con bajo ruido.

En este caso, al utilizar el PLL se añaden pequeñas fluctuaciones en la fase (phase jitter), produciendo un desfase instantáneo de unos pocos miligrados de ruido con respecto a la frecuencia de referencia. Este ruido es trasladado también a la salida X y a la salida Y dando lugar a la detección de señales cercana a la referencia.

En cambio, cuando se emplea una referencia interna, es decir no se suministra ninguna señal cuadrática de referencia para producir la sinusoidal, la señal producida por el oscilador interno se engancha a la fase interna, a bajas frecuencias, es decir frecuencias menores que 10Hz, este enganche es producido por el DSP, en caso contrario se emplea un comparador en fase discreto. En este caso resulta obvio, se prescindiría del uso del PLL. Por lo tanto, en este modo tampoco habría

las fluctuaciones de fase citadas anteriormente, por lo que se aconseja utilizar este modo cuando se necesite aplicaciones con ausencia de ruido de fase.

Como se ha citado anteriormente, el lockin tiene dos PSD que generan dos diferentes salidas, una llamada X que es la componente de cuadratura e Y la componente de fase. El oscilador interno genera por tanto una segunda señal sinusoidal desplazada θ_{REF} para el PSD encargado de generar la salida X y otra onda sinusoidal calculada por el DSP desplazada 90° a partir de la segunda onda.

Las ecuaciones de estas ondas serian:

Segunda onda sinusoidal: $\sin(w_{REF} + w_R t)$

Tercera onda sinusoidal: $\sin(w_{REF} + w_R t + \pi/2)$

En el caso del modelo que nos ocupa, la precisión de las ondas calculadas es de 20 bits en punto flotante.



Los detectores sensibles a la fase (PSD)

El funcionamiento del PSD, detector sensible de fase, es el de multiplicador lineal, multiplicando la onda de entrada por la onda de referencia.

En cuanto a tipos, se pueden clasificar en analógicos como en digitales:

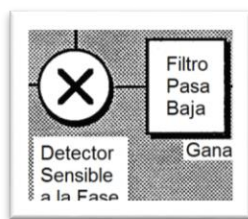
-Analógico: Multiplican ambas señales de manera analógica, esto es de manera no lineal y por tanto presentan numerosos problemas. Uno de ellos es el offset de salida, dado que la salida es una corriente continua el PSD contribuye al error y a una desviación del cero (zero drift). Esto es debido a la naturaleza de la multiplicación, existe el concepto de reserva dinámica (situado en torno a 60dB) que cuando existe una señal con un ruido elevado (medido en dB) superior a esta reserva dinámica, el PSD a la hora de multiplicar mide la señal con un error.

También cabe citar que los PSDs analógicos son susceptibles también en variaciones en la amplitud de la salida debido a la variación en la amplitud de la entrada de referencia.

-Digital: Los filtros digitales carecen de los problemas presentados por los detectores analógicos, esto es debido a que la multiplicación es extremadamente lineal y presenta una precisión de 20 bits y un muestro de 4us de la señal.

Para efectuar esta multiplicación es preciso incorporar un conversor A/D en la entrada del mismo, al fin de convertir tanto la onda de referencia como la onda de entrada en digital. En un lock-in digital la reserva dinámica queda limitada por la calidad de la conversión citada anteriormente, dado que una vez la entrada es digitalizada no se introducen errores a posteriori, no obstante citar que la reserva lineal es de 100dB.

Otro de los problemas eliminados por el PSD digital, es que este carece de error en la salida debido a que las fluctuaciones en la amplitud de la onda de referencia quedan suavizadas completamente.



Constantes de tiempo y ganancia en corriente continua

La salida del detector de fase contiene numerosas señales, la mayoría de estas señales presentan unas frecuencias que son sumas o diferencias entre la frecuencia de referencia y la frecuencia de la señal de entrada. Solamente aquella frecuencia de la señal de entrada que sea exactamente igual a la de referencia, dará lugar a una salida en corriente continua.

En la salida del PSD, se situará otro filtro pasa baja que va a eliminar todas las señales en corriente alterna indeseadas (es decir las señales que son suma y resta) como las componentes de ruido. Este filtro permite clasificar al lock-in como un excelente filtro de banda estrecha.

La constante de tiempo es un valor de configuración del lock-in que permite ajustar su ancho de banda de sus filtros pasa baja mediante la configuración del mismo, esta constante normalmente se configura con este valor: $\frac{1}{2\pi f_c}$, donde f_c es la frecuencia de corte [21] [22], la frecuencia a la cual la amplitud de la señal de entrada se atenúa en 3 dB [23]. Dado que la salida es una señal en corriente continua que es susceptible del ruido como se ha citado anteriormente, esta constante del tiempo es capaz de limitar o de eliminar completamente aumentando el valor de la misma.

Los filtros pasa-baja [24] [25, p. 37] del PSD citados, son en realidad simples filtros RC, con 6 dB/octava de pendiente roll-off. Habitualmente estos filtros se suelen concatenar dos de manera sucesiva de tal manera que presenten un roll-off de 6dB por octava a 12 dB por octava. La constante de tiempo se refiere al punto de 3 dB de atenuación de cada uno de ellos individualmente no del conjunto en sí.

El lock-in SR844 muestra la constante de tiempo y el ancho de banda equivalente de ruido (ENBW) en el menú de constante de tiempo. El ENBW no sería el polo a -3 dB del filtro, es sino el ancho de banda efectivo para el ruido gaussiano.

Volviendo sobre la comparativa entre PSDs analógicos y digitales, ahora se centrará en comparar los filtros pasa-baja que también poseen una vertiente digital. Conviene citar que el SR844 utiliza filtros digitales, mejorando a los analógicos logrando un total de 24 dB/Octava. Los filtros analógicos, no solo tienen un menor roll-off, sino que además necesitaría diversos mecanismos electrónicos de ajuste y operación para la constante de tiempo que incrementarían el lock-in.

Un incremento del roll-off a 24 dB/octava en cuatro etapas permite por ejemplo en el caso de tener una señal de referencia de 1 kHz y una fuerte señal de ruido a 1.05 kHz, las salidas obtenidas en el PSD, antes de pasar por la última etapa de filtros pasa-baja, de 250 kHz (suma) y 50 Hz (resta). En este caso al filtrar la señal de 50Hz, si la señal está 80dB por encima de la señal y se quisiese medir la señal al 1% entonces la señal componente necesita ser reducida 120 dB, hacer esto con un filtro de dos etapas conllevaría un retraso de 3 segundos, mientras que un filtro de cuatro etapas se reduciría a 100 ms.

Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama

Otra ventaja del filtrado digital es la habilidad de realizar un filtrado síncrono, esto es, si la señal de entrada está libre de ruido, la salida del PSD siempre va a contener una componente suma de ambas frecuencias, cuya amplitud va a igualar o superar la salida en continua en función de la fase. Para resolver este problema no se puede recurrir a la constante de tiempo, ya que sería demasiado grande.

Así pues, el filtro síncrono consistiría en promediar la salida obtenida por el detector de fase durante un ciclo completo de la frecuencia de referencia, esto lograría que todas las componentes que sean múltiplos de la referencia quedasen completamente eliminadas.

En el SR844 los filtros síncronos están disponibles para su uso, a frecuencias de detección inferiores a 200Hz. Para frecuencias mayores, los filtros serían innecesarios, para frecuencias mayores los filtros son innecesarios

Constantes de tiempos elevadas

Si la frecuencia de referencia es muy baja, es decir inferior a 1 Hz, y además existe una gran cantidad de ruido de baja frecuencia, la salida del detector contendrá muchas componentes de muy baja frecuencia. Para poder discriminarlas, es necesario utilizar constantes de tiempos muy grandes, en este caso y utilizando la formula $\frac{1}{2\pi f_c}$ se necesitaría una constante superior a 100 segundos, siendo esta totalmente prohibitiva. El SR844 ofrecería en cambio constantes de tiempo de hasta 30000 segundos a frecuencias de referencia inferiores a 200Hz.

Ganancia de salida

El valor de la salida en continua del PSD depende directamente de la reserva dinámica que se presentó anteriormente. Con 60 dB de reserva dinámica, la señal de ruido puede ser 1000 veces mayor que una señal de fondo de escala.

Para el detector, el ruido no puede superar su rango de entrada, en un lock-in analógico que soporte una amplitud de entrada de 5V y 60dB de reserva dinámica, la señal de salida sería de tan sólo 5 mV. Como amplificar esta señal de salida introduciría errores es preciso que la ganancia de salida en continua sea aproximadamente la misma que la reserva dinámica (1000).

El lock-in digital no tiene un amplificador en continua analógico, la ganancia de salida sigue siendo otra función controlada por el procesador digital. La amplificación consiste en tomar los números de entrada y multiplicarles por la ganancia deseada.

Reserva dinámica

En los apartados anteriores se ha abordado de manera superficial el concepto de reserva dinámica.

La reserva dinámica se define como el cociente entre la mayor señal de ruido tolerable respecto a la señal del fondo de escala, expresado como se ha venido haciendo en decibelios.

El concepto de tolerable es lo que se va comentar a continuación. El ruido en el límite de reserva dinámica no debería ser una sobrecarga en cualquier parte del lock-in (amplificador de entrada, PSDs, filtros pasa-baja o el amplificador DC), para regularlo se ajusta la distribución de la ganancia, aumentando la reserva para evitar sobrecarga en cualquiera de esas partes. En el caso del lockin digital no hay problema, pero en un lockin analógico aumentar la reserva dinámica provoca errores en la medición.

El problema más común es un error en la salida en continua, provocado por el ruido. Este último puede aparecer como un error de ganancia o un offset, dado que ambos defectos proceden de la frecuencia y amplitud del ruido, no podrían ser desplazados a cero en todos los casos limitando por

tanto la precisión de la medición. Al ser ruido en continua, estos errores no tendrían un carácter discreto sino continuo, por lo que aumentar la constante de tiempo no resultaría de utilidad. Por tanto, se puede definir como ruido tolerable, como aquel nivel de ruido que no afecta a la salida más allá de un pequeño porcentaje del fondo de escala.

Otra consecuencia de la reserva dinámica alta, es la generación de ruido y deriva a la salida. Esto es debido a que el amplificador de la señal continua funciona a ganancias muy altas y el ruido de baja frecuencia y la deriva del desplazamiento a la salida del PSD o la entrada del amplificador en continua se verán amplificados y aparecerán mayores en la salida. Este ruido presenta un grado de tolerabilidad mayor que el citado anteriormente, ya que un incremento en la constante de tiempo disminuirá este ruido. La deriva en corriente continua de un lockin analógico cuando se utiliza una reserva dinámica de 60 dB es de 1% del fondo de escala cuando varía en 10°C la temperatura en el experimento.

La reserva dinámica depende también de la frecuencia del ruido, es decir, el ruido a la frecuencia de referencia llegará a las salidas sin atenuación de modo que la reserva dinámica en este caso será de 0 dB. A medida que el ruido se aleja de la frecuencia de referencia, la reserva dinámica se incrementa, debido a que el filtro pasa-baja situado después del PSD atenúa este ruido. Por lo tanto la reserva dinámica depende de la constante de tiempo del filtro pasa-baja, y su ritmo de variación vendrá dado por el ritmo con el que se incremente la pendiente del roll-off del citado filtro, es por eso que los filtros de 24 dB/octava son mejores que los de 6 o 12 dB/octava. Cuando la frecuencia del ruido está muy alejada, la reserva estará limitada por la distribución de la ganancia y el nivel de sobrecarga de cada elemento de ganancia.

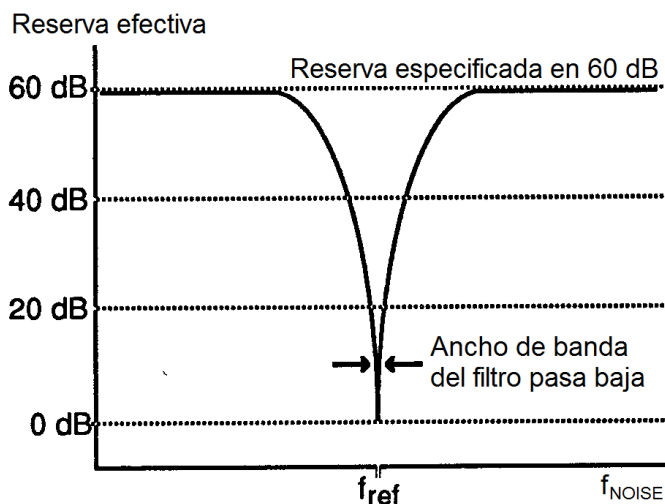


Ilustración 9- Representación de relación entre la frecuencia y la reserva del lock-in, extraído de [1]

El gráfico superior muestra la reserva real (también llamada efectiva) en comparación con la frecuencia del ruido (f_{noise}), la entrada de señal atenúa a frecuencias alejadas del rango de operación del lock-in mayores que 100 kHz. En estos casos la reserva podría ser mayor que en esas frecuencias del rango de operación.

No obstante, la supresión de ruido a frecuencias muy alejadas de la de referencia se escapa del interés del lockin.

Existe también el concepto de reserva mínima, entendida como una cantidad mínima de reserva dinámica. A ganancias elevadas, la reserva dinámica mínima se incrementa a partir de los 37 dB al

mismo ritmo que la sensibilidad se incrementa. Para un lock-in analógico, una reserva mínima inferior implica menor error de salida y deriva. En uno digital, el contrario no se produce, es decir una reserva mayor no implica mayor error de salida o deriva. No obstante, si la ganancia de la señal analógica previa al conversor digital es alta, el ruido que porta quedará amplificado a la salida a un mayor nivel de ruido que a la entrada. En este punto, el ruido detectado reflejará el propio ruido de la señal de entrada y no el amplificado por el conversor, por tanto, el hecho de incrementar la ganancia analógica reduciendo la reserva no reducirá el ruido de salida. Por tanto, no hay razón ninguna para reducir la reserva.

Si se establece la sensibilidad a un valor de $5\mu V$, la ganancia analógica es lo suficientemente alta como para que el ruido del conversor digital no suponga ningún problema. Para sensibilidades más bajas, no se debe introducir ganancia dado que no mejorará la relación señal ruido. El SR844 ante este último caso, incrementa la reserva mínima.

El SR844, con los PSD digitales, no sufre de errores de salida en continua provocados por ruidos fuertes. Su reserva dinámica puede incrementarse a más de 100 dB sin dar lugar a errores en la medición. Además, una ganancia elevada en continua no da lugar a un incremento en la deriva de salida. El único problema con altas reservas dinámicas ($> 60\text{dB}$) es el ruido producido por el conversor que se ve incrementado. Este incremento de ruido solo es visible cuando la reserva está por encima de 60 dB y por encima de la reserva mínima.

Por tanto, si el ruido no alcanza el límite de la reserva, el ruido de salida del SR844 podría ser detectable a reserva ultra altas, para corregir este caso bastaría con reducir la reserva dinámica y la ganancia en continua bajará y el ruido de salida también bajará.

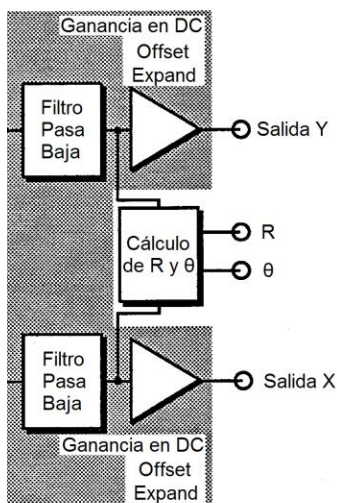
La dependencia de la frecuencia de la reserva dinámica es inherente a la técnica a la técnica de detección del lock-in. El SR844 al disponer de cuatro etapas de filtrado pasa baja, puede incrementar la reserva dinámica cerca de la frecuencia de referencia.

Resolución

Al igual que en el lock-in analógico, donde el ruido no puede exceder el rango de entrada del detector de fase, en el lock-in digital el ruido no puede superar el rango de entrada del conversor analógico digital.

Con un conversor de precisión de 16 bits, y una reserva dinámica de 60 dB, el ruido obtendría de rango los 16 bits completos, mientras que la señal utilizaría solo 6 bits. Aumentar esta reserva dinámica solamente perjudicaría el resultado, reduciéndose el número de bits necesarios en la salida.

Para solucionar esto, es necesario recurrir de nuevo al filtrado, en este caso a filtros pasa-baja situados en la salida del conversor. Estos filtros, como se ha citado anteriormente, tienen la capacidad de promediar numerosos datos con un cierto factor de retardo, así en la salida del mismo, se obtendría la señal promediada de los segundos previos (en función de la constante de tiempo utilizada) introducida en el conversor.



Salidas en corriente continua y escalado



Ilustración 10-Panel frontal y trasero de un amplificador lock-in

El lock-in SR844, posee dos salidas X e Y en su panel trasero y dos salidas Canal 1 y Canal 2 (CH1 e CH2) en su panel frontal.

X e Y son las salidas de los dos detectores de fase con el filtrado pasa baja, el desplazamiento (offset) y expansión (expand) ambas con un ancho de banda de salida de 100 kHz.

Las dos salidas del panel frontal (CH1 e CH2) se diferencian de las anteriores en que pueden ser configuradas para proporcionar salidas de X e Y, R y θ , o trazas 1-4.

En el caso de seleccionar como salida R y θ , los voltajes son proporcionales a los valores calculados por el lock-in para estos parámetros, estos cálculos se efectúan a una frecuencia de 512 Hz.

La sensibilidad del lock-in viene dada por el valor eficaz de la amplitud de un seno de entrada a la frecuencia de referencia, que da lugar a una salida en corriente continua a fondo de escala. Este fondo de escala implica 10V de valor en continua en las salidas X,Y o R, esto garantiza una ganancia global de 10V/sensibilidad. Esta ganancia se distribuye entre la ganancia en AC previa al detector de fase y la ganancia en continua posterior al mismo.

La modificación de la reserva dinámica a un valor determinado de dB, variará la distribución de esta ganancia, manteniendo inalterada la global.

El SR844 considera a 10V como valor de fondo de escala para X,Y o R, y por tanto de CH1 y CH2 cuando se configurar para obtener estos valores.

Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama

Los amplificadores lock-in están dispuestos de un mecanismo para calcular el valor RMS o valor eficaz de la señal de entrada en AC. Los valores mostrados en la pantalla que conforma el lock-in no serían por tanto valores absolutos, sino que serían valores RMS.

Por último, el valor de la fase es decir de θ , es un valor de rango $[-180^\circ, 180^\circ]$ que es independiente de la sensibilidad explicada. Cuando aparece por las salidas CH1 o CH2 un voltaje proporcional a θ , la escala o rango de salida es $18^\circ/\text{voltio}$ o lo que es lo mismo $180^\circ = 10\text{V}$.

Offset y expand

En la salida del lock-in aparece la opción de offset y de expand, ambas se detallan a continuación:

- **Offset:** Desplazamiento sobre el fondo de escala de las distintas salidas, que resulta de utilidad para medir desviaciones de la señal a partir de cierto valor. Se pueden programar desplazamientos en el siguiente rango: $[-100\%, 100\%]$
- **Expand:** Esta operación multiplica la señal de salida (sin contar el offset) por un factor de expansión. Esta expansión es útil para ampliar la resolución de medición en torno a algún valor cercano al cero. En el caso del lock-in SR844 se puede expandir la salida por un factor de 1 a 2^8 siempre y cuando la salida expandida no supere al fondo de escala. En el caso de un lock-in analógico, su salida con desplazamiento y expansión se corresponde a la siguiente fórmula:

$$Salida = \left(\frac{Señal}{Sensibilidad} - Desplazamiento \right) * 10\text{ V} * Expansión$$

Ecuación 8- Salida del lock-in contando con la operación de expand

Donde el desplazamiento es un valor inferior a 1 ($50\% = 0.5$), la expansión es un entero de 1 a 256.

Por último, citar que las funciones de desplazamiento y expansión de X e Y en el SR844 son funciones de salida que no afectan al cálculo de R o θ . R tendría su propio desplazamiento y expansión de salida.

Ruido

En los siguientes apartados se explicarán las distintas fuentes de ruido con las que el lock-in se puede encontrar durante su operación.

Ruido térmico (o ruido Johnson)

Las resistencias generan un voltaje de ruido entre sus terminales debido a la variación térmica en la densidad de la de densidad de carga de la propia resistencia (por eso al ruido Johnson se le conoce también como ruido térmico) [26]. El voltaje producido en un circuito abierto se describe por la siguiente ecuación:

$$V_R(rms) = \sqrt{4kTR\Delta f}$$

Ecuación 9- Ruido térmico

Donde k es la constante de Boltzmann ($1.38 * 10^{23} \frac{J}{K}$), T es la temperatura en grados Kelvin, R es el valor de la resistencia en Ohmios, y Δf es el ancho de banda durante la medición.

Ruido rosa

Adicional al ruido Johnson existe un ruido adicional presente en las resistencias conocido como ruido rosa [27]. Estos ruidos provienen de las fluctuaciones en la resistencia debida a las corrientes que lo atraviesan. Para resistencias hechas de carbono este ruido está entre 0.1 y 3 micro voltios rms por cada voltio aplicado a la resistencia. Al contrario, las resistencias de película metálica y las resistencias de hilo bobinado tienen en torno a 10 veces menos ruido. Este ruido tiene un espectro de $\frac{1}{f}$ y complica las mediciones a baja frecuencia.

Este ruido rosa está también en válvulas al vacío y semiconductores.

Ruido de disparo (o ruido impulsivo)

El ruido impulsivo o de disparo es el ruido producido por la propia corriente eléctrica debido a la naturaleza finita de los portadores de carga, es decir, el ruido debido a la variación en el flujo de electrones en la corriente [28]. Este ruido puede aparecer al medir la corriente o al pasar a través de una resistencia. La ecuación de este ruido viene dada por:

$$I_R(rms) = \sqrt{2qI\Delta f}$$

Ecuación 10- Ruido de disparo

Donde q es la carga del electrón ($1.6 * 10^{-19}$) Culombios, I es la corriente RMS en AC o corriente en DC dependiendo del circuito, y Δf es el ancho de banda. Cuando la corriente de entrada del lock-in es para medir la corriente de una señal en AC, el ancho de banda es tan pequeño que el ruido del disparo no es relevante.

Ruido total

El ruido aleatorio total es la raíz cuadrada de la suma de los cuadrados de todas las fuentes de ruido anteriores.

Fuentes de ruido externas

Hasta ahora se han explicado las fuentes de ruido del lock-in, pero existen fuentes de ruido externa. Estas fuentes de ruido son asíncronas, es decir, no están relacionadas con la señal de referencia, ni su frecuencia ni sus armónicos. Por poner ejemplos de estas fuentes, pueden ser, radios, pantallas

de tubo, equipos de refrigeración, motores, luminarias... Para contrarrestar los efectos que puedan tener estas fuentes de ruido

Sin embargo, algunas fuentes de ruido están en relación con la referencia, y si entran a formar parte de la señal, se sumarán o restarán a la auténtica señal y provocarán errores en la medición.

Acoplamiento capacitivo

Un voltaje en AC proveniente de algún aparato eléctrico cercano puede producir capacidades parasitas. A pesar de que esta capacidad, $C_{PARASITA}$, puede ser muy pequeña el ruido, que genere puede superar fácilmente a la señal en medición. Esto resulta especialmente perjudicial si el ruido acoplado va a la misma frecuencia que la de referencia

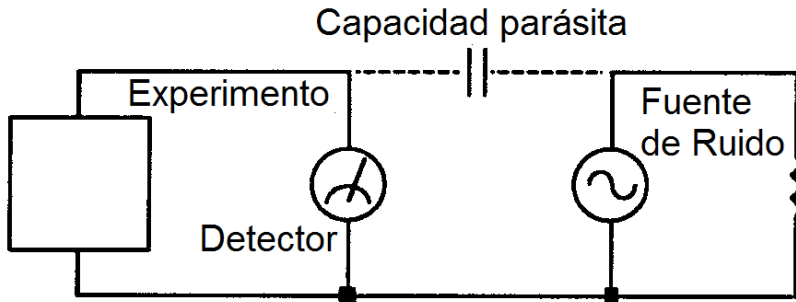


Ilustración 11-Esquema del experimento de acoplamiento capacitivo, que incluye al ruido

Se puede calcular, la corriente de ruido provocada por la capacidad parásita mediante la siguiente formula:

$$i = C_{PARASITA} \frac{dV}{dt} = \omega C_{PARASITA} V_{RUIDO}$$

Ecuación 11- Corriente parásita

Donde ω es 2π veces la frecuencia del ruido, V_{RUIDO} es la amplitud del ruido y $C_{PARASITA}$ es la capacidad parásita expresada en faradios.

Si la fuente de ruido es síncrona con el experimento, es decir, tiene la misma frecuencia de trabajo que la de referencia el problema de la medición se agrava, dado que el lock-in lo interpreta como si fuera una señal.

Algunas soluciones para esto son las siguientes:

- Mantener la fuente de ruido lejos del experimento.
- Formular el experimento para medir voltajes con impedancia baja
- Instalar un apantallamiento colocando el experimento en una caja de metal.
- Apagar la fuente de ruido

Acoplamiento inductivo.

Una corriente alterna, en un aparato cercano puede acoplarse con el experimento a través de un campo magnético. Al existir una corriente variable en un circuito cercano, se da, como consecuencia a un campo magnético variable que induce una f.e.m. $\frac{d\theta_B}{dt}$ en el lazo que conecta el

detector con el experimento. Esto es similar a un transformador en el que el lazo experimento-detector funciona como el devanado secundario.

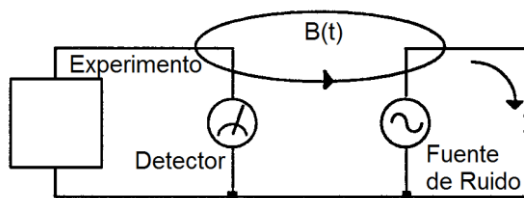


Ilustración 12-Esquema del experimento de acoplamiento inductivo

Para poder evitar este problema, se puede recurrir a las siguientes soluciones:

- Medir las corrientes, usando detectores de alta impedancia.
- Apagar la fuente de ruido.
- Reducir el área del lazo, utilizando cables coaxiales.
- Utilizar apantallamiento magnético para impedir que el campo magnético atravesase el área del experimento.

Acoplamiento resistivo

Se refiere a aquellas corrientes que al atravesar las conexiones de tierra pueden dar lugar a voltajes de ruido. Al igual que en anteriores casos, si estas corrientes de tierra trabajan a la frecuencia de referencia dan lugar a errores en la medición.

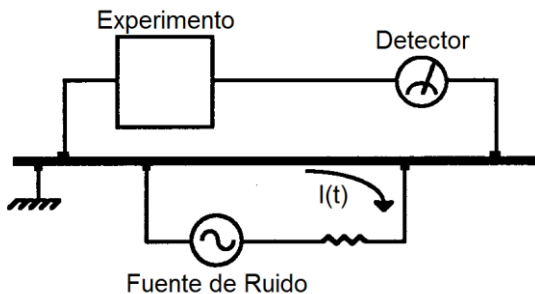


Ilustración 13-Esquema del experimento de acoplamiento resistivo

En la figura presentada, el detector está midiendo la señal relativa a una tierra que está ubicada lejos del resto del experimento. El lock-in detectará tanto la señal del detector y el voltaje causado por el retorno de tierra. Ambos el detector como la fuente de ruido están conectados a tierra en distintos lugares también con un potencial diferente.

Para solucionar este problema se tienen las siguientes soluciones:

- Separar las fuentes de tierra grandes de aquellas que sean empleadas para señales pequeñas.
- Que todas las conexiones de tierra estén conectadas al mismo punto físico.
- Utilizar una barra de puesta a tierra gruesa para reducir las resistencias de las conexiones a tierra.

Efectos termopares

Otra de las posibles fuentes de ruido es debida a la fuerza electromotriz creada por las uniones entre metales diferentes.

Este ruido es de baja frecuencia, ya que la temperatura del detector y del experimento varía lentamente. Este efecto sin embargo es apreciable en el rango de salida de muchos detectores y puede suponer por tanto un problema en mediciones de baja frecuencia, en especial para el rango de mHz.

Para mitigar estos efectos termopares son:

- Utilizar una segunda unión en polaridad inversa lo que genera una segunda fuerza electromotriz que cancela el potencial térmico de la primera unión.
- Mantener constante la temperatura del experimento.

Efectos microfónicos

El ruido mecánico puede convertirse en eléctrico mediante los efectos microfónicos. Cambios accidentales de naturaleza física en el experimento pueden dar lugar a ruido eléctrico en distintas frecuencias del lock-in.

A modo de ejemplo, si un cable coaxial que esté conectado a un detector con un lock-in. La capacidad eléctrica del cable está en función de su geometría. Las vibraciones mecánicas del cable producen una variación en el tiempo de la capacidad, dado que el cable está regido por

$$i = \frac{dQ}{dt} = C \frac{dV}{dt} + V \frac{dC}{dt}$$

Ecuación 12- Efectos microfónicos

Las vibraciones mecánicas en el cable provocan un $\frac{dC}{dt}$ que producirá una corriente en el cable. Esta corriente afecta al detector y a la señal medida.

Para mitigar los efectos de las señales microfónicas se utilizan los siguientes métodos:

- Utilizar un cable de bajo ruido diseñado para tal.
- Inmovilizar los cables que transmitan señales al lock-in
- Eliminar las vibraciones mecánicas de los aparatos que componen el experimento.

Mediciones de ruido

Otra de las utilidades de un lock-in es que pueden utilizarse para realizar mediciones de ruido. Lo que es de gran utilidad en el desarrollo de nuevos dispositivos.

El SR844 mide el ruido de la señal de entrada a la frecuencia de referencia. En concreto las señales de entrada que pertenecen al ancho de banda de detección, marcado por la constante de tiempo y el roll off del filtro pasa baja, aparecen a la salida de frecuencia, el ruido de entrada [29, pp. 27,30] próximo a la frecuencia de referencia aparece como ruido a la salida con un ancho de banda que va desde corriente continua hasta el ancho de banda de detección.

El ruido puede definirse como la desviación estándar (la raíz cuadrada de la media de las desviaciones al cuadrado) de las salidas, R, X o Y. El SR844 puede medir este ruido con exactitud, almacenando primero el valor de salida en una tabla para luego calcular la desviación estándar usando funciones aplicadas. Este ruido, es la desviación estándar dividida por la raíz cuadrada del ancho de banda equivalente de ruido de la constante de tiempo, y se mide en $\frac{V}{\sqrt{Hz}}$.

El ruido medido mediante el uso de la desviación estándar, no permite ofrecer una salida en tiempo real, por lo que el SR844 dispone de otro método para estimar el ruido directamente.

Para ello, se establece una traza X_n , Y_n o R_n , valores calculados en tiempo real y que son una estimación del ruido de cada uno de los canales.

Cada uno de ellos se calcula usando el siguiente algoritmo. El valor medio actual del canal se resta al valor actual para hallar la desviación de ese canal respecto a su media. Como último paso se calcula media móvil del valor absoluto de las desviaciones, a esta media se le denomina MAD (mean average deviation). El MAD y el RMS difieren salvo que el ruido sea gaussiano en cuyo caso están relacionados por un factor constante.

El SR844 utiliza el método MAD para estimar los valores de ruido RMS, X_n , Y_n y R_n . La ventaja de esta técnica es la velocidad, el tiempo de cálculo depende de la constante de tiempo. El SR844 selecciona el tiempo de promediado y varía entre 10 veces y 80 veces el valor de la constante de tiempo. Reducir la constante de tiempo para mejorar el tiempo de promediado, aumenta el ancho de banda permitiendo un mayor nivel de ruido, al contrario, aumentar promueven valores más estables, pero tarda más tiempo en proporcionarlo.

Ruido de R

Si X e Y presentan un ruido equivalente y están centradas en tornos al cero, entonces los valores de R serán siempre positivos y con un valor distinto de cero. En este caso, el ruido de X e Y da a lugar a una media R que puede ser considerada como el mínimo detectable de R. El cálculo del ruido de R por cualquier método, hará que sus valores sean siempre menores que X e Y, esto es debido a que X e Y tienen valores que producen desviaciones grandes, mientras que R es siempre positiva y con media distinta de cero, por lo que su desviación será pequeña. Por tanto, el ruido producido por R no es indicativo del ruido gaussiano que se mide.

Por último, si el valor de R está en estado estable y es distinto de cero, con desviaciones debidas al ruido pequeñas comparadas con la media de R, entonces el valor de R tiene sentido, sería el caso de cuando realmente se puede detectar ruido dentro de una señal.

2.3.3 Usos del amplificador Lock-in

En este apartado, se expondrán algunos usos de interés del amplificador Lock-in, este apartado ha sido extraído en gran parte de [1].

La mayoría de usos del lock-in están ligados a técnicas de física, química y electrónica.

Detectores de gas

En el artículo de De Marcellis et al. [30], se utiliza un amplificador Lock-in para detectores de gas de tipo resistivo, basado en resistencias, consiguiendo un aumento del 80% en la sensibilidad del sistema.

Espectroscopia de fluorescencia

Los procesadores digitales de señal se encuentran en numerosos instrumentos de laboratorio de química, dichos procesadores llevan incorporado un Lock-in. Un ejemplo es el espectrofotómetro de fluorescencia, en los que un haz de luz, normalmente ultravioleta, irradia una muestra de una sustancia. Las moléculas absorben entonces la energía de los fotones que inciden en ellas, pasando a un estado electrónico excitado. A continuación, las moléculas vuelven a alguno de los niveles vibratorios emitiendo un fotón con una energía equivalente a la del salto. Al haber distintos niveles, se emitirán fotones de diferentes longitudes de onda.

El haz de fotones emitido por la muestra pasa por un monocromador, que transforma el haz en diferentes haces de distintas frecuencias, pudiendo seleccionar mediante una ranura móvil uno de estos componentes. De esta forma manteniendo la frecuencia de excitación de la muestra constante, es posible obtener un barrido de las frecuencias de emisión. Aplicar un lock-in a este tipo de espectroscopia proporciona numerosas ventajas.

Espectroscopia de impedancia electroquímica

Esta técnica sirve para realizar reacciones de corrosión sobre un material dado, en general un metal. El proceso consiste en montar una celda electroquímica con tres electrodos, llamados electrodo de referencia, de trabajo, y auxiliar. Posteriormente un instrumento llamado potenciostato, genera una corriente alterna que atraviesa el electrodo de trabajo, fabricado con el material de estudio, y mide su respuesta en intensidad de corriente.

Finalmente, el Lock-in, conectado al experimento, se encargará de sintetizar la frecuencia y de calcular el desfase en la señal recibida, así como de la impedancia.

Resonancia paramagnética electrónica

La resonancia paramagnética electrónica, es una técnica que permite detectar especies químicas con electrones desapareados, tales como compuestos paramagnéticos como son la mayoría de los metales de transición [31]. Es decir, permite detectar la presencia de impurezas en muy bajas concentraciones, incluso llegando al nivel atómico, lo que garantiza la fabricación de metales con más calidad.

El proceso, consiste en colocar la muestra de análisis en un resonador de microondas refrigerado por helio, situado en el campo magnético producido por dos bobinas de Helmholtz, en una de las cuales hay una sonda de efecto Hall. Finalmente, a la sonda se encuentran conectados dos electrodos, procedentes de la cámara de resonancia, se hace pasar un voltaje a través de ellos y se generan pulsos de microondas en la cámara de resonancia, a una frecuencia dada. Para medir la corriente de la muestra, se utiliza un amplificador de transimpedancia, la frecuencia de referencia a la que se emiten los microondas y la señal de corriente procedente de la muestra pasa a un amplificador Lock-in, que calcula la amplitud y el desfase producido por la muestra, para esa frecuencia.



Ilustración 14- Espectrómetro EPR en la Universidad Jagelónica, Polonia. Fotografía de Przemyslaw "Tukan", extraída de [32]

Resonancia magnética nuclear

En la resonancia magnética nuclear, los amplificadores Lock-in también tienen aplicación. En un experimento típico, la frecuencia de referencia suministrada por el lock-in entra hacia la sonda donde se coloca la muestra. De esta forma es posible medir, utilizando la frecuencia de salida de la sonda, las variaciones en la amplitud y en la fase. Un artículo en el que se menta explícitamente una aplicación del Lock-in en resonancia magnética nuclear es el de Kan.



Ilustración 15-Montaje de un espectrómetro de resonancia magnetica nuclear (NMR) en el Laboratorio de Experimentación [33]

Balanzas

A la hora de medir masas, se han utilizado como instrumento las galgas extensiométricas, que están formadas por un circuito en forma de zigzag realizado en un material conductor al que le atraviesa una corriente. Debido al efecto piezorresistivo, la resistencia del material cambia cuando se deforma, al colocar este dispositivo a lo largo de un brazo metálico suspendido por un extremo por una plataforma y con un platillo en el otro, es posible medir la flexión del brazo al colocar un objeto encima del platillo. El amplificador lock-in mediría la conductancia a través de la galga.

Sin embargo, en el artículo de Abul Al Aish et al. [34], se presenta un modelo de balanza con sensores de carácter capacitivo en lugar de resistivos. El diseño de los mismos consiste en tres cilindros concéntricos de latón, donde dos de ellos están conectados entre sí formando un condensador. Entre medias y sin hacer contacto está el tercer cilindro, soldado al platillo, conectado a masa y suspendido por un muelle. De esta forma, al colocar un peso en el platillo, el tercer cilindro se desplaza hacia abajo, modificando la capacidad del condensador creado por los otros cilindros. Para obtener la componente en cuadratura de la corriente generada habría que seguir el esquema presentado a continuación:

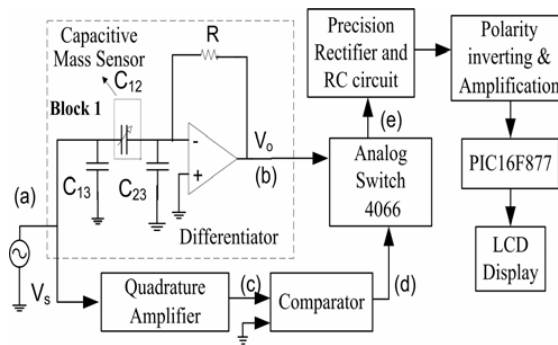


Ilustración 16- Diagrama de bloques del sensor capacitivo de masa

En esta figura, se utiliza el microcontrolador PIC16F877 para procesar la componente en corriente continua de la señal en cuadratura.

Microscopía de fuerza atómica

La microscopía de fuerza atómica, sirve para detectar irregularidades sobre la superficie de una muestra.

Su funcionamiento es muy sencillo, se basa en una lámina de metal flexible, en cuyos extremos hay una punta afilada. Esta lámina, conocida como micropalanca, se conecta a un elemento piezoeléctrico que puede hacer vibrar la lámina. La muestra se coloca debajo de la punta, y un diodo láser emite un haz de luz que al chocar con la lámina refleja la luz y es recogida por un fotodetector, cuando la lámina se mueve debido a las irregularidades de la muestra, el ángulo de reflexión cambia y el haz modificando la luz recibida por el fotodiodo.

La siguiente figura ilustra el funcionamiento del mismo:

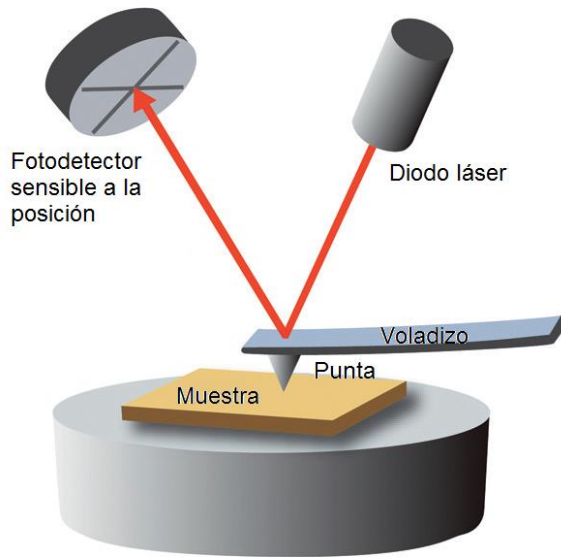


Ilustración 17- Diagrama de funcionamiento de un microscopio de fuerza atómica, extraído de [35]

Esta técnica sin embargo es muy sensible al ruido provocado por las vibraciones, por lo que es conveniente montar los instrumentos sobre mesas estabilizadas de óptica.

Al vibrar la lámina como consecuencia de las variaciones detectadas en la superficie, las vibraciones presentan una determinada frecuencia de resonancia, el uso de amplificadores Lock-in permitiría mejorar las imágenes provenientes de este tipo de técnica. En concreto, la señal de referencia sería la resonancia de la lámina, y por el canal de salida del sistema se conectaría la salida de la matriz de fotodetectores, mejorando sensiblemente la calidad de la imagen.

Existe una variante del sistema, llamada microscopía de fuerza atómica bimodal [36] que hace resonar la lámina a dos frecuencias a la vez, siendo una de ellas la frecuencia fundamental y otra un armónico de esta. En este caso se utilizarían dos amplificadores Lock-in diferentes, uno para cada una de las dos frecuencias de resonancia sobre las que se extraen la amplitud y el desfase de la señal de salida del sistema.

Medición del sonido

En el siguiente experimento [37], se utiliza un amplificador lock-in para medir la velocidad del sonido:



Ilustración 18- Fotografía con el montaje para medir la velocidad del sonido usando lock-in, captura de [37]

En este caso un generador de frecuencias se conecta a una de las entradas del lock-in (la señal de referencia), mientras que a la otra se conecta un micrófono con dos diodos led (uno rojo, y otro azul) en su superficie. El generador de frecuencias, se conecta también a un altavoz que traduce la señal sinusoidal en una señal positiva.

Al acercar el micrófono al altavoz, debido al acoplamiento, ambas señales entrarán en fase, mientras que al alejarse o variar el grado la fase cambiará, cuando la fase de la señal de entrada coincida con la del micrófono, se encenderá el led rojo, mientras que cuando esté la fase en torno los 90° , se encenderá el led azul.

Variando la frecuencia y la fase recogida por el lock-in al acercar o alejar el micrófono sobre una regla, es factible calcular la velocidad del sonido, teniendo en cuenta el desplazamiento del micrófono sobre la regla.

3. Capítulo tercero: GESTIÓN DEL PROYECTO

El propósito de este capítulo y de los próximos es de comentar la metodología desde el punto de vista de la Ingeniería del Software aplicada al desarrollo del proyecto.

En este capítulo se van a exponer los siguientes apartados:

- Definiciones.
- Alcance del proyecto.
- Participantes o stakeholders del proyecto.
- Costes del proyecto.
- Planificación del proyecto.
- Gestión del proyecto.

3.1 Definiciones

- **Ingeniería del Software:** La ingeniería del software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza. [38]
- **Desarrollo de un sistema software:** El desarrollo de un sistema software constituye un proyecto que está compuesto por un número de actividades, cada actividad a su vez está formada por una colección de tareas; una tarea consume recursos y produce resultados. Un resultado puede ser un sistema, un modelo o un documento. Los recursos pueden ser participantes tiempo o equipamiento [38].
- **Producto Software:** Un producto software es un sistema software que se acompaña de la documentación necesaria para su instalación e uso [39]
- **Stakeholders:** Se denominan stakeholders a los participantes, o lo que es lo mismo, a todas aquellas personas vinculadas al proyecto pudiendo ser:
 - Clientes
 - Desarrolladores
 - Gestores del proyecto.
- **Roles:** Conjunto de tareas asignadas a un participante. [39]
- **Modelo:** Representación abstracta del sistema que nos permite responder a preguntas acerca de dicho sistema.
- **Proceso:** Es el conjunto de actividades [38] que conllevan la creación de un producto software. Estas actividades se basan en modelos, que proporcionan una visión abstracta del proceso. Existen tres modelos de proceso común que son:
 - **Modelo de desarrollo Iterativo:** El desarrollo iterativo se basa en la idea de desarrollar una implementación inicial exponiéndola a los comentarios del usuario y refinándola a través de las diferentes versiones hasta que se desarrolla un sistema adecuado.
 - **Modelo en cascada [40]:** El modelo en cascada se basa en la ejecución de 5 actividades una detrás de otra no pudiéndose empezar la siguiente sin haber completado la anterior.

- **Ingeniería del software basada en componentes:** En la ingeniería de software basada en componentes se basa en la reutilización de componentes software de manera masiva.
- **Artefacto:** Cualquier resultado obtenido a lo largo del proceso del desarrollo del software. Puede formar parte del producto final, ser un prototipo o bien distintos resultados de cualquier tarea del proyecto [38].
- **Dominio del problema:** Es la descripción de los aspectos del sistema del mundo real que se consideran relevantes para el proyecto, es decir aquellos aspectos del mundo real que ofrecen una comprensión del entorno [39].
- **Requisito:** Especificación de alto nivel, normalmente en lenguaje natural, sobre lo que debería hacer el sistema [41].
- **Arquitectura:** Descripción de la organización, motivación y estructura de un sistema software [39].
- **Open UP:** OpenUP es un método y proceso de desarrollo software perteneciente a la compañía Eclipse. Se basa en tres axiomas [42]
 - Está dirigido por requisitos, casos de uso y riesgos.
 - Centrado en la arquitectura.
 - Iterativo e Incremental.

Open UP divide sus tareas en cinco etapas:

- **Gestión del Proyecto:** Etapa en la que se establecen los tiempos y recursos de los que va a constar el proyecto, en OpenUP dura todo el tiempo del proyecto.
- **Análisis o licitación de requisitos:** Se basa en obtener lo que espera el cliente de la aplicación analizando para ello los requisitos del sistema y depurarlos mediante casos de uso.
- **Diseño del sistema:** Se basa en establecer la arquitectura del sistema y su división en clases para su posterior implementación.
- **Implementación**
- **Pruebas**

OpenUP se basa en iteraciones, que son planificadas en un intervalo definido de tiempo que no superan las pocas semanas.

OpenUP estructura el ciclo de vida de un proyecto en cuatro fases: concepción, elaboración, construcción y transición.

OpenUP divide el desarrollo de un proyecto en cuatro fases fundamentales:

- **Inicio:** Define el ámbito del proyecto y desarrolla los casos de negocio.
- **Elaboración:** Se elabora el Plan del Proyecto, se especifican las características y el marco de referencia de la arquitectura.
- **Construcción:** Se construye el proyecto
- **Transición:** Se pasa el producto a los usuarios.
- **SCRUM:** Scrum [43] es el nombre con el que se denomina a los marcos de desarrollo ágiles caracterizados por:
 - Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.

- Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos auto organizados, que en la calidad de los procesos empleados.
- Solapamiento de las diferentes fases del desarrollo, en lugar de realizar una tras otra en un ciclo secuencial o de cascada.
- **Kanban [44]:** Kanban es un método para gestionar el trabajo intelectual, con énfasis en la entrega justo a tiempo, mientras no se sobrecargan a los miembros del equipo. En este enfoque, el proceso, desde la definición de una tarea hasta su entrega al cliente, se muestra para que los participantes lo vean y los miembros del equipo tomen el trabajo de una cola.

3.2. Perspectiva y objetivo del proyecto

El objetivo de este proyecto es el de desarrollar una aplicación software de control y simulación de generador de señal y osciloscopio, que permita a su vez la creación de un lock-in de bajo coste que pueda ser utilizado en el departamento, este último objetivo, es el tratado por mi compañero Eduardo Rodríguez Gutiez en su trabajo final de máster [1].

3.3. Participantes o Stakeholders del proyecto

La gestión y planificación del proyecto es necesaria para las siguientes personas:

1. El jefe de proyecto, para planificar el horario del proyecto y los recursos necesarios.
2. Los miembros del equipo del proyecto, para entender lo que ellos necesitan hacer, cuando necesitan hacerlo y ver otras actividades de las que dependen.

El rol de jefe de proyecto, recae sobre los tutores, Helena y Salvador, mientras que los roles de miembros del equipo del proyecto recaen sobre Andrés Escobar Cotán y Eduardo Rodríguez Gutiez, desarrolladores de distintos TFM pero que comparten una línea común.

3.4. Costes del proyecto

A continuación, se especificarán los costes estimados del proyecto presentado, en términos de costes de hardware, de software y de personal.

En el caso de costes de hardware, se han tenido en cuenta el ordenador comprado para el desarrollo de este proyecto, así como el del osciloscopio, el pre-amplificador de señal y el lock-in para comparar.

No obstante, para ejecutar la aplicación es necesario un PC con los siguientes requisitos hardware:

- o Procesador Intel o AMD de 1 GHz o más, se recomienda PCs con doble núcleo. No funciona en Netbooks con procesadores VIA y Intel Atom.
- o 1 GB de RAM o más
- o Tarjeta Gráfica de más 64 MB de memoria RAM
- o Espacio libre en el disco duro de más de 10 GB

A continuación, se realiza una tabla con costes promedios extraídos de la tienda PC BOX, los precios son a fecha de la redacción de este documento.

Costes Hardware	
PC: Ordenador Imaz i5-4460	694,23 €
Osciloscopio / Generador de señal: Keysight DSO-X 3104 A	11642,00 €
Interfaz USB GPIB: Interfaz USB GPIB Keysight 82357B [45]	687,28 €
Preamplificador: SRS SR445A	1182,10 €
Lock-In: Amplificador Lock-In SRS SR844	8324,00 €
Total Costes	22529,61 €

Tabla 1- Costes Hardware

Costes Software

En el caso de los costes software, la mayoría de las librerías utilizadas son gratuitas debido a acuerdos con la Universidad, sin embargo, de cara a una analogía con un proyecto real del mercado

Costes Software	
Visual Studio Ultimate 2012	13299,00€
Matlab R2013b	69,00€
Team Foundation Server 2012	93,26€

Bonobo Git Server	0€
USB Monitor Pro	152,60€
Keysight IO Libraries Suite	0€
BenchLink Waveform Builder Pro	690,00€
Enterprise Architect 12.0	212,70€
Balsamiq Mockups 3.1	79,70€
Microsoft Office Professional Plus 2016	438,90€
Total Costes	13.738 €

Tabla 2- Costes Software

Costes de personal

En el caso de costes de personal se ha supuesto que se cobra por proyecto entregado, y no por horas. Las personas que han trabajado en el proyecto son las comentadas anteriormente.

La relación de personas y costes es la que viene a continuación:

Costes Personal	
Jefe de proyecto	2000 €
Analista	1800 €
Diseñador	1500 €
Programador	1300 €
Ingeniero de Pruebas	1300 €
TOTAL COSTES	7900 €

Tabla 3- Costes de personal

Por lo tanto, el coste total del proyecto será:

$$\text{Coste: } 226,95 \text{ €} + 94,30 \text{ €} + 7900 \text{ €} = 8221,25 \text{ €}$$

3.5. Planificación del proyecto

El desarrollo del proyecto será dirigido usando el modelo de proceso OpenUP [42], explicado anteriormente, ya que utiliza UML como lenguaje de modelado y se basa en un modelo de ciclo de vida dirigido por los casos de uso, centrado en la arquitectura e iterativo e incremental.

Dentro de las fases especificadas en la definición de OpenUP, se pueden realizar diferentes iteraciones de carácter incremental con el fin de profundizar en las actividades desarrolladas en dicha fase. Cada iteración finaliza con una visión del artefacto o artefactos que se vayan a desarrollar en la citada fase.

En este caso, cada iteración corresponderá con cada una de las fases marcadas con hitos en el plan del proyecto.

La calendarización del proyecto es una actividad que distribuye estimaciones de esfuerzo a través de la duración planificada del proyecto. Esta actividad se representa mediante gráficos, que muestran las dependencias entre las diferentes actividades que conforman el proyecto.

3.5.1. Iteraciones en el proyecto.

El proyecto llevado a cabo se ha planteado como cuatro iteraciones:

Primera iteración (1 de octubre – 20 de octubre)

En esta primera etapa, consistió en un estudio del proyecto a realizar, en términos de conceptos teóricos.

A posteriori se pasó al análisis de requisitos correspondiente y al estudio de las tecnologías y dispositivos solicitados.

Se comenzó desarrollando la aplicación en LabView, sin embargo, por desconocimiento de la plataforma, y por falta de completitud de la misma respecto al proyecto planteado, se descarto.

Posteriormente se procedió a estudiar la posibilidad de usar MatLAB, herramienta de desarrollo conocida por el equipo y más cercano a un paradigma de programación de alto nivel que LabView.

Sin embargo, los drivers del generador Agilent ofrecían soporte limitado, eliminando la posibilidad de utilizarlo de generador de señal.

Finalmente, consultando las propiedades de los distintos drivers del osciloscopio/generador, se observó que ofrecían wrappers más completos para C# y .NET.

Dado que un miembro del equipo conocía la plataforma se optó finalmente por esta plataforma como marco de desarrollo.

Segunda iteración (21 de octubre- 17 de noviembre)

En esta segunda interacción una vez definida la plataforma y efectuado una licitación preliminar de requisitos, se decide elaborar un prototipo de la aplicación final para estimar problemas que puedan surgir a posteriori y medir posibles errores.

Tercera iteración (18 de noviembre – 10 de diciembre de 2015)

En esta iteración se decide refactorizar el prototipo presentado en la anterior fase y por lo tanto reconstruir la aplicación añadiendo lo que falta.

Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama

El día 14 de diciembre, entre a trabajar parte de GMV y comencé un máster titulado Máster GAIA in Professional Development 4.0 [46], ofrecido por la Universidad de Alcalá de Henares. En un principio, la jornada laboral era de 8,45 horas por lo que me vi obligado a reducir el tiempo empleado en el proyecto a partir de esta fecha.

Sábados y domingos se dedicaron a elaborar los trabajos requeridos en el Máster citado anteriormente que tenían un carácter semanal.

Aunque se intentó dedicar dos horas al día al proyecto, esto no pudo ser así debido a los riesgos 6 y 7 citados en el siguiente apartado.

Cuarta iteración (10 de diciembre de 2015 – abril 2016)

En esta última iteración se introducen algoritmos de mejora de la suavidad de las ondas.

Y se mejora la adaptabilidad y el soporte a nuevos módulos que se puedan añadir en el futuro.

También se procede a redactar esta memoria.

3.5.2. Riesgos

El objetivo de todo proyecto debe ser un producto con la mayor calidad y minimizar el impacto de los riesgos que se puedan representar [38]

Un riesgo conlleva dos características:

- Probabilidad
- Pérdida

En el ámbito de la Ingeniería del Software, el riesgo es una variable que perjudica al éxito de un proyecto.

Categorías de riesgos

Los riesgos se pueden clasificar en diferentes categorías. En concreto existen seis categorías en las que se puede etiquetar un riesgo:

- Riesgos tecnológicos.
- Riesgos de personas
- Riesgos organizacionales
- Riesgos relacionados con las herramientas.
- Riesgos relacionados con los requisitos.
- Riesgos de estimación.

Ante un riesgo es necesario no solo una monitorización del mismo, sino definir una estrategia para reaccionar cuando este se produzca, en concreto las estrategias más comunes son:

- Impedir el riesgo.
- Transferir el riesgo.
- Aceptar el riesgo.
- Mitigar el riesgo.
- Aceptar un plan de contingencia.

Tabla de riesgos

Para este proyecto se han identificado los siguientes riesgos:

[RG-001]: Los requisitos capturados no son los correctos.	
Tipo:	Riesgo de requisitos
Probabilidad:	Alta
Impacto:	Catastrófico. Aumenta el tiempo de desarrollo del proyecto.
Plan de Contingencia:	Reunirse con los jefes de proyecto / clientes tantas veces como sea necesaria, repetir el análisis de requisitos y planificar otra vez los tiempos del proyecto.

Tabla 4 - Riesgo 001

[RG-002]: Falta de comprensión del proyecto.	
Tipo:	Riesgo de estimación.
Probabilidad:	Alta
Impacto:	Alto. Aumenta el tiempo de desarrollo del proyecto.
Plan de Contingencia:	Reunirse con los jefes de proyecto / clientes para clarificar las dudas planteadas.

Tabla 5- Riesgo 002

[RG-003]: Mal diseño del proyecto.	
Tipo:	Riesgo de estimación.
Probabilidad:	Baja
Impacto:	Medio. Aumenta el tiempo de desarrollo del proyecto.

Plan de Contingencia:	Identificar el problema del mal diseño. Estimar en tiempo su corrección, aumentar si es necesario las horas dedicadas al proyecto. Corregir el defecto encontrado.
------------------------------	---

Tabla 6 - Riesgo 003

[RG-004]: Error en la implementación.	
Tipo:	Riesgo tecnológico.
Probabilidad:	Muy alta
Impacto:	Baja. Aumenta el tiempo de desarrollo del proyecto.
Plan de Contingencia:	Identificar el problema en la implementación. Estimar en tiempo su corrección, aumentar si es necesario las horas dedicadas al proyecto. Corregir el defecto encontrado.

Tabla 7- Riesgo 004

[RG-005]: Modificación de los requisitos.	
Tipo:	Riesgo de requisitos
Probabilidad:	Media
Impacto:	Bajo.
Plan de Contingencia:	Identificar en recursos de tiempo, el diseño e implementación de dicho recurso. Debido a la naturaleza ágil de la metodología utilizada, es una característica deseable.

Tabla 8 - Riesgo 005

[RG-006]: Incremento de tasa de trabajo en GMV.	
Tipo:	Riesgo organizacional
Probabilidad:	Muy alta
Impacto:	Muy alta. Invalida el tiempo planificado para el proyecto.
Plan de Contingencia:	Aceptar el riesgo es la primera solución posible. Mitigación: Solicitar algún día de vacaciones para dedicarlo al proyecto

Tabla 9 - Riesgo 006

[RG-007]: Incremento de tasa de trabajo en el Máster GAIA.	
Tipo:	Riesgo organizacional
Probabilidad:	Muy alta.
Impacto:	Muy alta. Invalida el tiempo planificado para el proyecto.
Plan de Contingencia:	Aceptar el riesgo es la única solución posible.

Tabla 10 - Riesgo 007

3.5.2.1. Monitorización de riesgos

En esta sección se describen los riesgos detectados durante el proyecto.

El primer riesgo no ha sucedido a lo largo del proyecto.

El riesgo 2º ocurrió en la segunda iteración, en concreto en las semanas 43 a 45 de 2014.

En concreto, el problema detectado fue que se pensó en resolver el problema del lock-in de recuperación de la onda original y cálculo de su desfase mediante el uso de Fourier.

Para mitigarlo se concertó diferentes reuniones con el tutor con el fin de encontrar otra solución al problema planteado.

El riesgo 3º no sucedió.

El riesgo 4º se produjo a lo largo de la 3ª iteración, en paralelo con los riesgos 6 y 7. Se detectó que la generación de señal en el dispositivo Agilent no se correspondía por lo introducido por la

Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama

aplicación. La solución fue la de realizar ingeniería inversa hasta averiguar el algoritmo usado internamente por el propio dispositivo para dibujar y generar ondas.

El riesgo 5º no sucedió en el proyecto.

El riesgo 6º ocurrió durante los siguientes intervalos de semanas, debido a que fue necesario dedicar más horas al proyecto que estaba asignado en la empresa:

- Semana 10-12 de 2015.
- Semana 18-24 de 2015.
- Semana 40-42 de 2015.

Para mitigarlos se escogieron cuatro lunes como días de vacaciones para trabajar en el proyecto, en concreto:

- 13,20,27 de Abril.
- 4 de Mayo.

Desafortunadamente, hubo días no contemplados en los que este riesgo de manera imprevisible sucedió, y cuya única opción fue aceptar el riesgo.

El riesgo 7º ocurrió de manera previsible durante la duración del Máster.

3.5.3. Calendario del proyecto

El calendario del proyecto es el presentado a continuación:

Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
Iteración 1	14 días	mié 01/10/14	lun 20/10/14	
Estudio viabilidad del proyecto	3 días	mié 01/10/14	vie 03/10/14	
Definición de requisitos	3 días	lun 06/10/14	mié 08/10/14	2
Estudio posibilidades LabView	2 días	jue 09/10/14	vie 10/10/14	3
Estudio Posibilidades MatLab	3 días	lun 13/10/14	mié 15/10/14	4
Estudio posibilidades C#	3 días	jue 16/10/14	lun 20/10/14	5
Tecnología Seleccionada	0 días	mié 01/10/14	lun 20/10/14	6FF
Iteración 2	21 días	mar 21/10/14	mar 18/11/14	7
Control del osciloscopio	3 días	mar 21/10/14	jue 23/10/14	
Dibujado de las señales recibidas	5 días	vie 24/10/14	jue 30/10/14	9
Generador de ondas compuestas	12 días	vie 31/10/14	lun 17/11/14	10

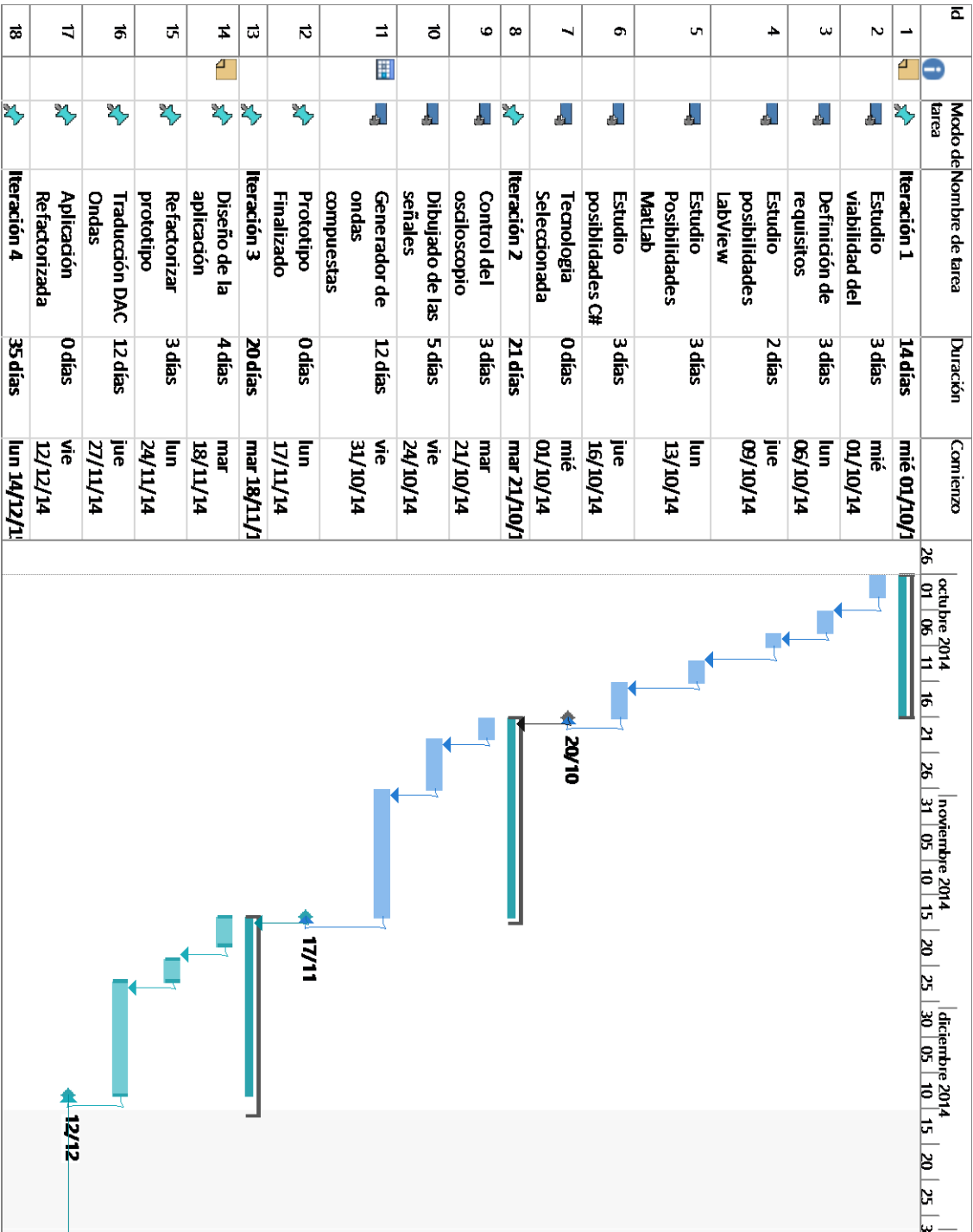
Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama

Prototipo Finalizado	0 días	lun 17/11/14	lun 17/11/14	11FF
Iteración 3	20 días	mar 18/11/14	lun 15/12/14	12
Diseño de la aplicación	4 días	mar 18/11/14	vie 21/11/14	
Refactorizar prototipo	3 días	lun 24/11/14	mié 26/11/14	14
Traducción DAC Ondas	12 días	jue 27/11/14	vie 12/12/14	15
Aplicación Refactorizada	0 días	vie 12/12/14	vie 12/12/14	16FF
Iteración 4	35 días	lun 14/12/15	lun 18/04/16	17
Redactar la memoria	35 días	lun 14/12/15	lun 18/04/16	
Memoria Finalizada	0 días	lun 15/02/16	lun 18/04/16	19FF
Algoritmos de suavizado	10 días	lun 14/12/15	lun 11/01/16	
Adaptar simulador	7 días	mar 12/01/16	mié 20/01/16	21
Aplicación Finalizada	0 días	mié 20/01/16	mié 20/01/16	22FF

Tabla 11 - Calendario del proyecto

Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama

A continuación, el diagrama de Gantt que representa gráficamente este calendario:



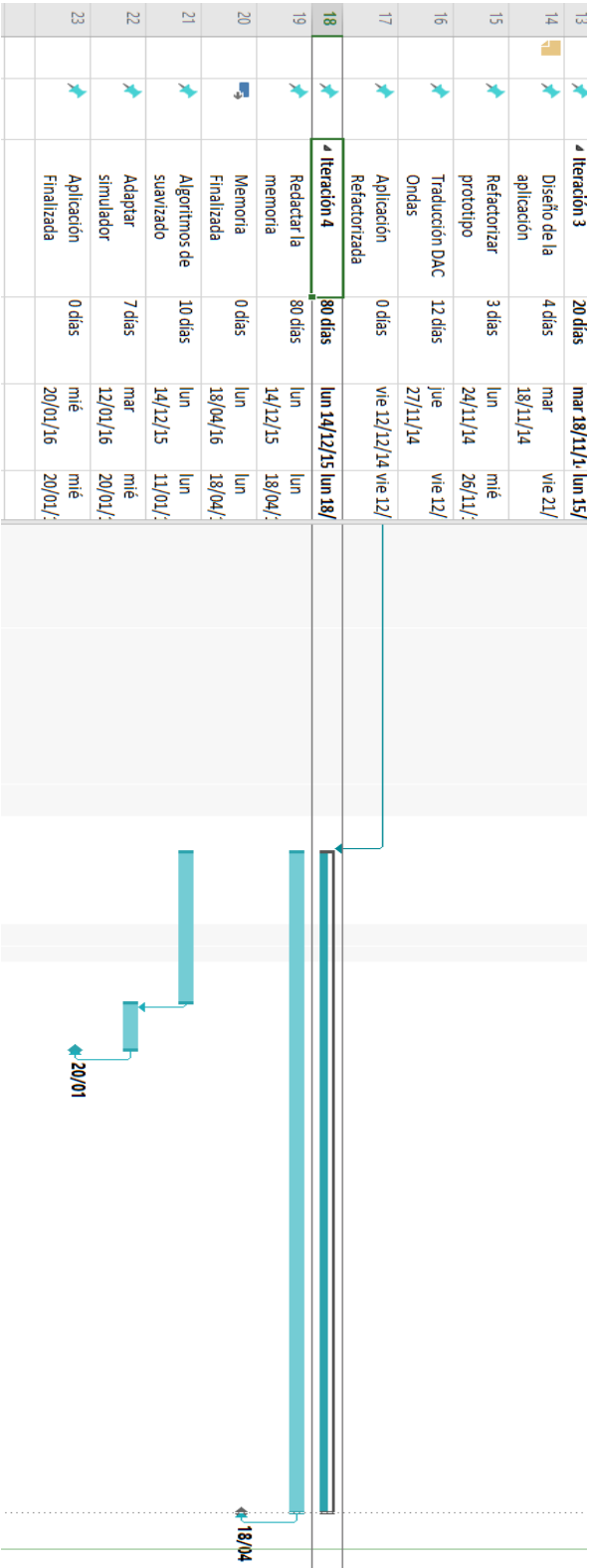


Ilustración 19 - Diagrama de Gantt del calendario del proyecto

4. Capítulo cuarto: ANÁLISIS DE LA APLICACIÓN

Este capítulo aborda la fase de análisis de un proyecto software desde el punto de vista del modelo en cascada, incorporando además de la gestión de requisitos, los casos de uso y diagramas de secuencia característicos del modelo Open UP.

Los artefactos definidos por Open UP son los siguientes:

- Definición de requisitos.
- Descripción de los casos de uso.
- Diagrama de secuencia de análisis de los casos de uso.
- Diagrama de clases de análisis.

4.1. Glosario

El glosario, necesario en la etapa análisis, es la parte del documento donde se especifican aquellos términos que se consideran de interés para el lector y cuya comprensión facilita la lectura de este capítulo. Algunos de estos términos [9] aparecen ya reflejados en el capítulo segundo de la memoria, salvo que aquí se le da un matiz menos técnico, de cara a comprender mejor el producto.

Circuito: Red eléctrica que se desea, medir, excitar mediante una corriente o bien simular.

Onda: Fenómeno vibratorio que transmite corriente eléctrica variando esta a lo largo del tiempo

Forma de onda: Dibujo o características matemáticas de la onda que se desea medir. Puede ser de los siguientes tipos:

- **Cuadrada**
- **Triangular**
- **Sinusoidal**
- **Rampa**
- **Pulso**
- **Ascenso cardinal**
- **Descenso cardinal**
- **Ruido aleatorio**
- **Arbitraria**
- **Continua**

Osciloscopio: Dispositivo que permite el visualizado de señales en un circuito.

Canal: Una de las entradas posibles del osciloscopio proveniente del circuito, también puede ser simulada.

Generador: Dispositivo que permite generar corriente según una determinada forma de onda.

Lock-In [1]: Dispositivo que amplifica una señal, normalmente muy pequeña y de alta frecuencia, camuflada en una señal mayor y que permite la obtención de sus características en especial fase y cuadratura.

4.2. Actores

Los actores, definen los distintos roles que interactúan con el sistema.

ACT-0001	Usuario
Versión	1.0 (17/10/2015)
Descripción	Este actor representa a un usuario tipo del sistema

Tabla 12- Actor Usuario

ACT-0002	Osciloscopio
Versión	1.0 (17/10/2015)
Descripción	Este actor representa a un osciloscopio físico que interactúa con el sistema, enviando y recibiendo datos.

Tabla 13- Actor Osciloscopio

4.3. Análisis de requisitos

En esta sección se describen los requisitos que conforman el sistema, se dividirán en funcionales, aquellos que especifican el qué del sistema y por lo tanto son de obligado cumplimiento, y los no funcionales, aquellos que especifican el cómo [39].

Requisitos Funcionales

FRQ-0001	Conexión y control de generador de ondas.
Versión	1.0 (11/10/2015)
Descripción	El sistema deberá poder conectarse y controlar el generador de ondas del Agilent DSO-3104A.

Tabla 14 - Requisito Funcional 001

FRQ-0002	Conexión a osciloscopio.
Versión	1.0 (11/10/2015)
Descripción	El sistema deberá <i>poder conectarse al osciloscopio Agilent DSO-3104A.</i>

Tabla 15 - Requisito Funcional 002

FRQ-0003	Obtención y muestra de canales
Versión	1.0 (11/10/2015)
Descripción	El sistema deberá <i>ser capaz de detectar y mostrar los canales analógicos que tenga el osciloscopio Agilent DSO 3104A.</i>

Tabla 16 - Requisito Funcional 003

FRQ-0004	Tiempo de muestreo
Versión	1.0 (11/10/2015)
Descripción	El sistema deberá <i>ser capaz de mostrar en tiempo real la gráfica del canal obtenida a través del osciloscopio cuando esta cambie a lo largo del tiempo.</i>

Tabla 17 - Requisito Funcional 004

FRQ-0005	Escala de canal
Versión	1.0 (11/10/2015)
Descripción	El sistema deberá <i>ser capaz de mostrar para el canal seleccionado el valor de escala de división del eje X y del eje Y.</i>

Tabla 18 - Requisito Funcional 005

FRQ-0006	Escala de canal
-----------------	------------------------

Versión	1.0 (11/10/2015)
Descripción	El sistema deberá <i>ser capaz de mostrar para el canal seleccionado el valor de escala de división del eje X y del eje Y.</i>

Tabla 19 -Requisito Funcional 006

FRQ-0007	Mezclador de ondas
Versión	1.0 (11/10/2015)
Descripción	<p>El sistema deberá <i>ser capaz de procesar señales sinusoidales simples y suma de estas mediante un mezclador de ondas.</i></p> <p><i>El mezclador de ondas podrá:</i></p> <ul style="list-style-type: none"> - <i>Crear ondas sinusoidales a partir de la entrada de pares valor de amplitud y frecuencia del usuario.</i> - <i>Modificar una onda sinusoidal creada previamente.</i> - <i>Eliminar una onda sinusoidal creada previamente.</i> - <i>Sumar entre si n ondas sinusoidales creadas previamente.</i>

Tabla 20 - Requisito Funcional 007

FRQ-0008	Mezclador de ondas, persistencia
Versión	1.0 (11/10/2015)
Descripción	El sistema deberá <i>ser capaz de almacenar y recuperar en ficheros ondas sinusoidales creadas previamente.</i>

Tabla 21- Requisito Funcional 008

FRQ-0009	Mezclador de ondas, generador
Versión	1.0 (11/10/2015)

Descripción	El sistema deberá <i>ser capaz de enviar la onda creada por el mezclador de ondas al generador del osciloscopio Agilent DSO 3104^a.</i>
--------------------	---

Tabla 22- - Requisito Funcional 009

FRQ-0010	Simulador de osciloscopio
Versión	1.0 (11/10/2015)
Descripción	<p>El sistema deberá <i>tener un simulador de osciloscopio, con 4 canales que muestre la siguiente información:</i></p> <ul style="list-style-type: none"> - <i>Canal 1: Onda proveniente del mezclador de ondas</i> - <i>Canal 2: La misma onda con el doble de frecuencia.</i> - <i>Canal 3: La misma onda con el triple de frecuencia.</i> - <i>Canal 4: La misma onda con el cuádruple de frecuencia.</i>

Tabla 23- Requisito Funcional 010

FRQ-0011	Fallo de conexión del osciloscopio
Versión	1.0 (11/10/2015)
Descripción	El sistema deberá cargar por defecto el simulador en caso de no poder conectar con el osciloscopio Agilent DSO 3104 ^a .

Tabla 24- Requisito Funcional 011

FRQ-0012	Componente en fase
Versión	1.0 (11/10/2015)
Descripción	El sistema deberá <i>ser capaz de extraer la componente de fase, a partir de los valores obtenidos en la medición</i>

Tabla 25- - Requisito Funcional 012

FRQ-0013	Componente en cuadratura
Versión	1.0 (11/10/2015)
Descripción	El sistema deberá <i>capaz de extraer la componente de cuadratura, a partir de los valores obtenidos en la medición</i>

Tabla 26- Requisito Funcional 013

FRQ-0014	Conductancia
Versión	1.0 (11/10/2015)
Descripción	El sistema deberá <i>capaz de extraer la conductancia, a partir de los valores obtenidos en la medición</i>

Tabla 27- Requisito Funcional 014

FRQ-0015	Capacitancia
Versión	1.0 (11/10/2015)
Descripción	El sistema deberá <i>ser capaz de extraer la capacitancia, a partir de los valores obtenidos en la medición</i>

Tabla 28-- Requisito Funcional 015

FRQ-0016	Persistencia
Versión	1.0 (11/10/2015)
Descripción	El sistema deberá <i>ser capaz de almacenar y recuperar los datos obtenidos de las ondas, tanto del osciloscopio como del generador.</i>

Tabla 29-- Requisito Funcional 016

Requisitos No Funcionales

NFR-0001	Osciloscopio como generador de señal
Versión	1.0 (13/10/2011)
Descripción	El sistema deberá <i>poder conectarse al osciloscopio Agilent DSO-3104^a y utilizarlo como generador de señal</i>

Tabla 30- Requisito No Funcional 001

NFR-0002	Osciloscopio como conversor
Versión	1.0 (13/10/2011)
Descripción	El sistema deberá <i>poder conectarse al osciloscopio Agilent DSO-3104^a y utilizarlo como conversor</i>

Tabla 31- Requisito No Funcional 002

NFR-0003	Lenguaje de programación
Versión	1.0 (13/10/2011)
Descripción	El sistema deberá <i>implementarse utilizando C# bajo .NET Framework 4.5</i>

Tabla 32- Requisito No Funcional 003

NFR-0004	Sistema Operativo
Versión	1.0 (13/10/2011)
Descripción	El sistema deberá <i>funcionar en Windows 7 como mínimo</i>

Tabla 33- Requisito No Funcional 004

NFR-0005	XML como estructura de almacenamiento
Versión	1.0 (13/10/2011)
Descripción	El sistema deberá <i>guardar los datos de las mediciones en un formato XML determinado.</i>

Tabla 34- Requisito No Funcional 005

4.4. Casos de Uso

Los casos de uso son descripciones en lenguaje natural del funcionamiento del sistema formulada en base a escenarios y usuarios.

El diagrama de casos de uso para esta aplicación es el que se manifiesta a continuación:

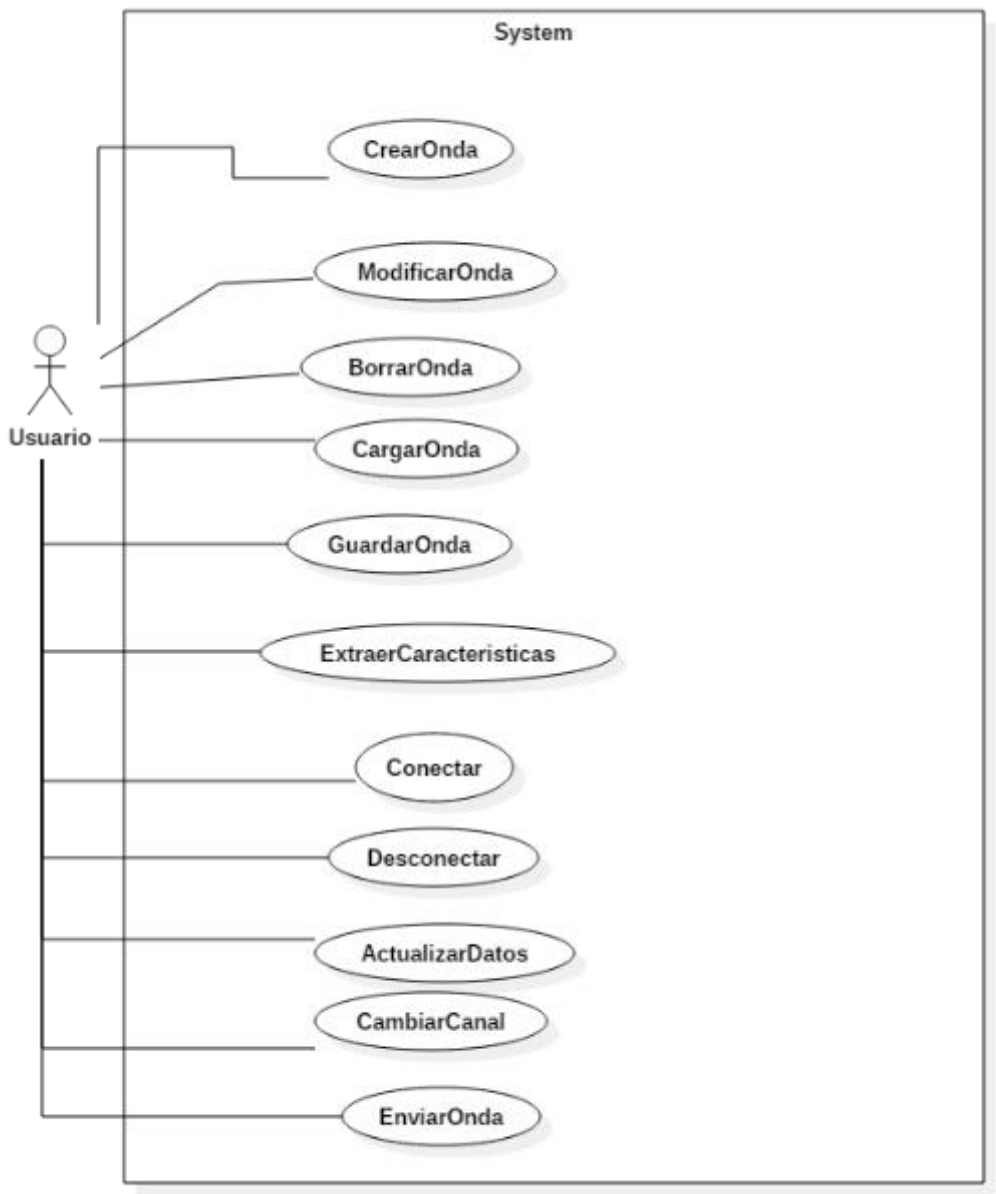


Ilustración 20- Diagrama de Casos de Uso

UC-001: Crear Onda

UC-0001	Crear Onda	
Versión	1.0 (13/10/2015)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desee crear una onda</i>	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El actor <u>Usuario(ACT-0001)</u> selecciona <i>Crear Onda</i>
	2	El sistema <i>solicita la información del usuario de la onda a crear</i>
	3	El actor <u>Usuario(ACT-0001)</u> introduce la <i>información de la onda a crear</i>
	4	El sistema <i>comprueba los valores introducidos</i>
	5	El sistema <i>realiza las operaciones necesarias para calcular la onda.</i>
	6	El sistema <i>muestra la onda creada al usuario.</i>
Postcondición	La nueva onda queda almacenada en el sistema	
Excepciones	Paso	Acción
	4	Si el Actor Usuario, ha introducido una información errónea, el sistema invalida los datos introducidos y a continuación el caso de uso queda sin efecto.

Tabla 35- Descripción de análisis del caso de uso 001

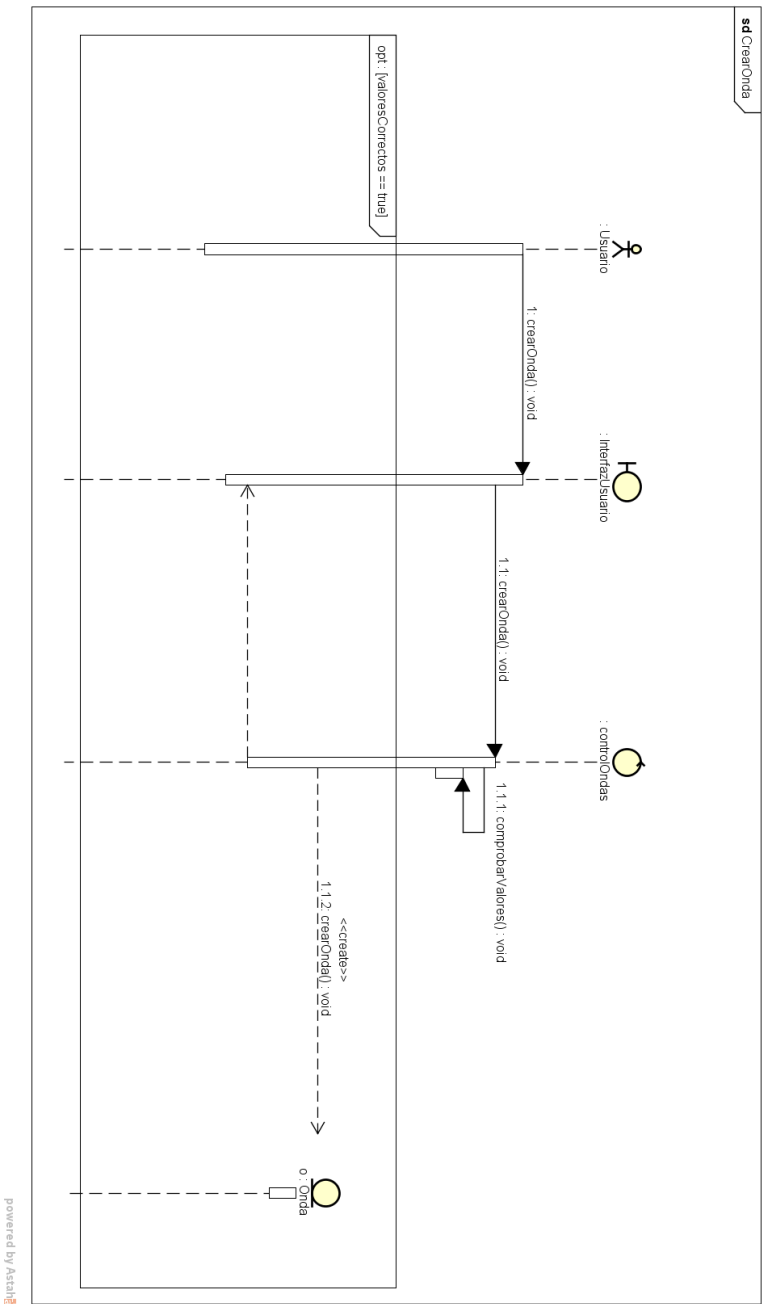
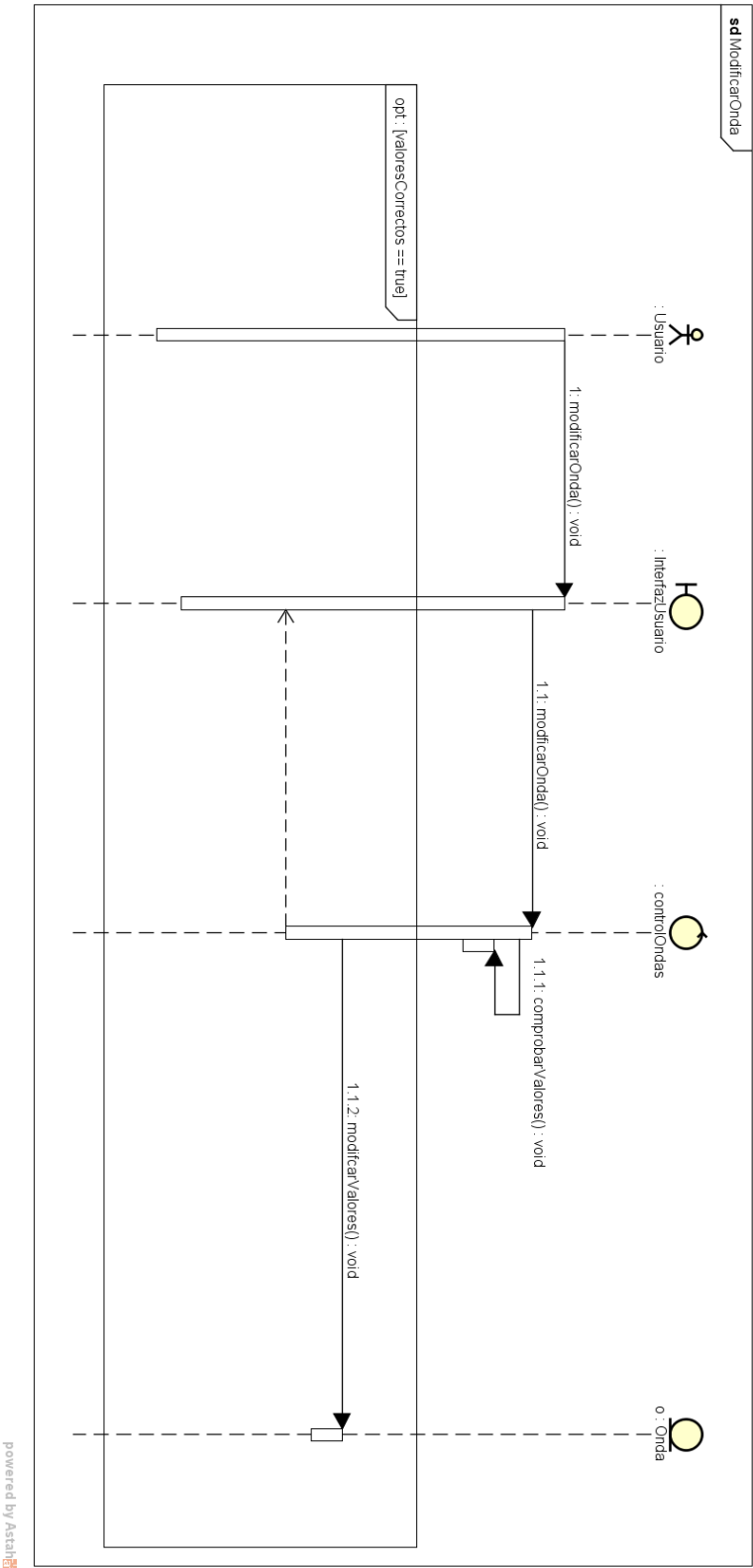


Ilustración 21- Diagrama de secuencia de análisis del caso de uso 001

UC-002: Modificar Onda

UC-0002	Modificar Onda	
Versión	1.0 (13/10/2015)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desee modificar una onda</i>	
Precondición	Que haya una onda cargada en el sistema	
Secuencia normal	Paso	Acción
	1	El actor <u>Usuario(ACT-0001)</u> <i>selecciona Modificar Onda</i>
	2	El sistema <i>solicita la información del usuario de la onda a modificar de la lista disponible</i>
	3	El actor <u>Usuario(ACT-0001)</u> <i>selecciona la onda a modificar</i>
	4	El sistema <i>solicita la información a modificar de la onda seleccionada</i>
	5	El actor <u>Usuario(ACT-0001)</u> <i>introduce la nueva información de la onda a modificar</i>
	6	El sistema <i>valida la información introducida.</i>
	7	El sistema <i>calcula la nueva onda.</i>
	8	El sistema <i>muestra la onda al usuario</i>
Postcondición	La onda modificada queda almacenada en el sistema	
Excepciones	Paso	Acción
	6	<i>Si el Actor Usuario, ha introducido una información errónea, el sistema invalida los datos introducidos y a continuación el caso de uso queda sin efecto.</i>

Tabla 36- Descripción de análisis del caso de uso 002



UC-003: Borrar Onda

UC-0003	BorrarOnda	
Versión	1.0 (13/10/2015)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desee borrar una onda</i>	
Precondición	Que haya una onda cargada en el sistema	
Secuencia normal	Paso	Acción
	1	El actor <u>Usuario(ACT-0001)</u> <i>selecciona BorrarOnda</i>
	2	El sistema <i>solicita la información del usuario de la onda a eliminar de la lista disponible</i>
	3	El actor <u>Usuario(ACT-0001)</u> <i>selecciona la onda a eliminar</i>
	4	El sistema <i>elimina la onda seleccionada.</i>
Postcondición	La onda queda eliminada del sistema.	
Excepciones	Paso	Acción

Tabla 37- Descripción de análisis del caso de uso 003

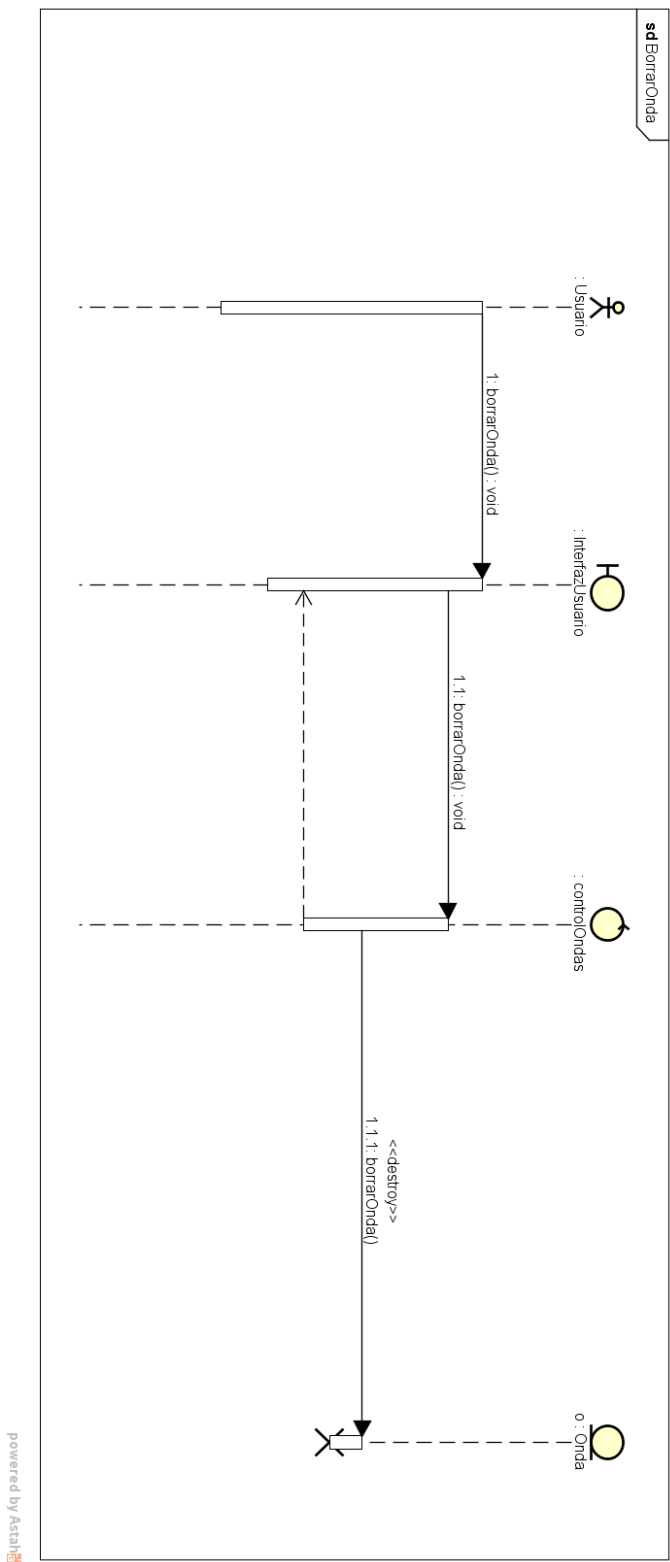
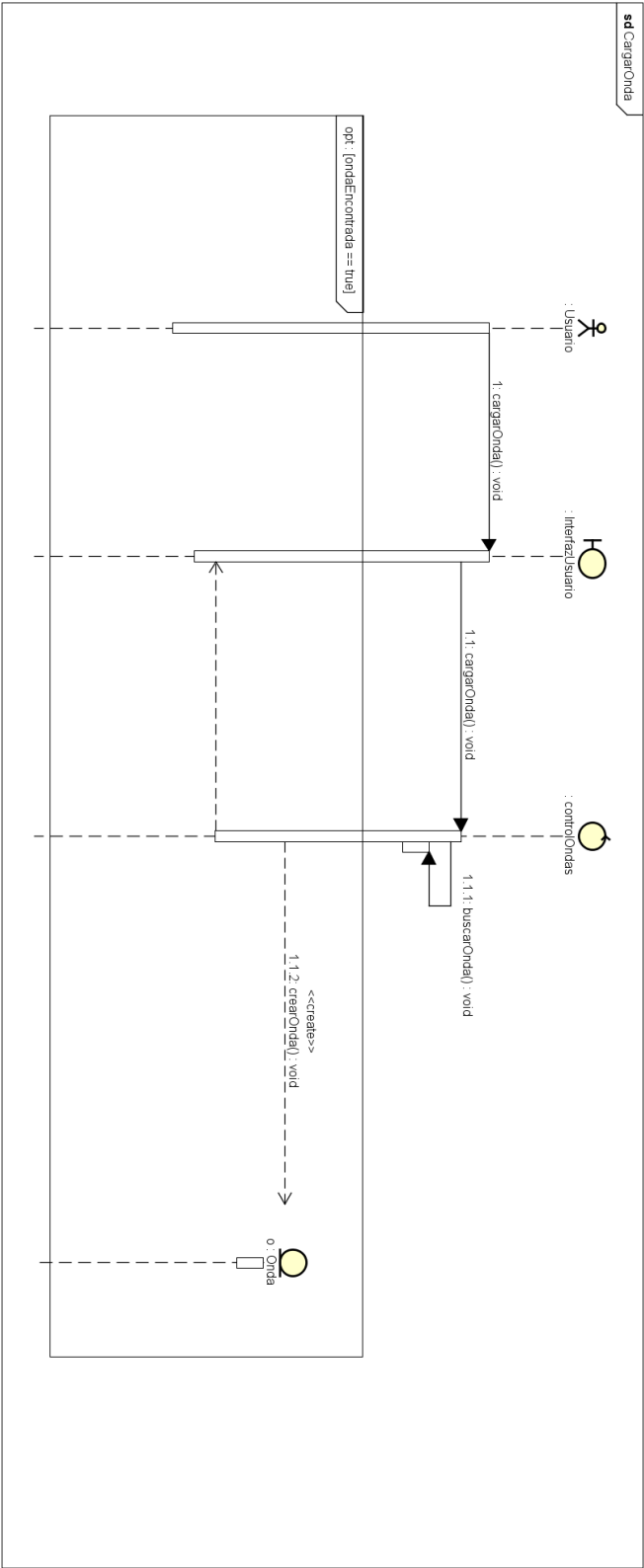


Ilustración 23-Diagrama de secuencia de análisis del caso de uso 003

UC-004: Cargar Onda

UC-0004	CargarOnda	
Versión	1.0 (13/10/2015)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desee cargar una onda</i>	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El actor <u>Usuario(ACT-0001)</u> <i>selecciona CargarOnda</i>
	2	El sistema <i>solicita la información del usuario de la onda a cargar</i>
	3	El actor <u>Usuario(ACT-0001)</u> <i>introduce la ruta de la onda a cargar</i>
	4	El sistema <i>procesa la onda a cargar</i>
	5	El sistema <i>muestra la onda cargada</i>
Postcondición	La onda queda cargada en el sistema	
Excepciones	Paso	Acción
	4	Si el sistema <i>no es capaz de cargar la onda correctamente, notifica al Usuario, a continuación el caso de uso queda sin efecto</i>

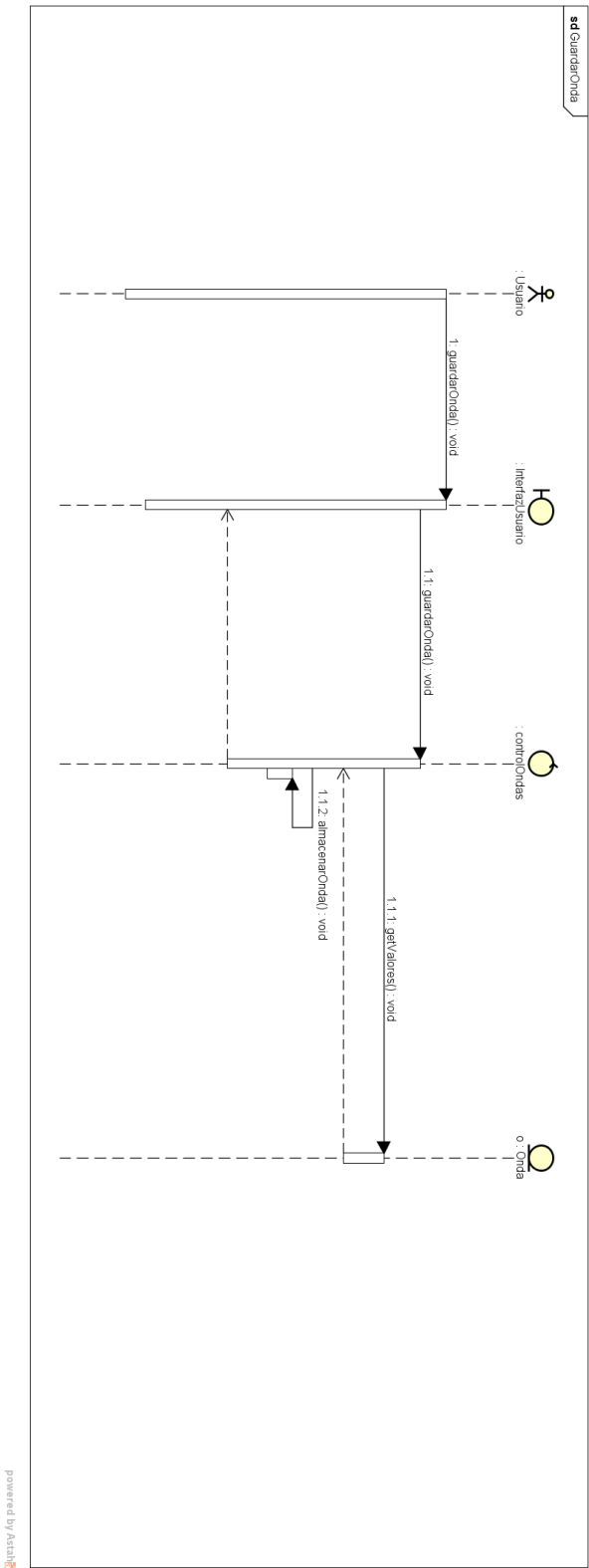
Tabla 38- Descripción de análisis del caso de uso 004



UC-005: Guardar Onda

UC-0005	GuardarOnda	
Versión	1.0 (13/10/2015)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desee guardar una onda</i>	
Precondición	Que haya una onda cargada en el sistema	
Secuencia normal	Paso	Acción
	1	El actor <u>Usuario(ACT-0001)</u> <i>selecciona GuardarOnda</i>
	2	El sistema <i>solicita la información del usuario de la onda a guardar de la lista disponible</i>
	3	El actor <u>Usuario(ACT-0001)</u> <i>selecciona la onda a guardar</i>
	4	El sistema <i>almacena la onda</i>
Postcondición	La onda queda guardada en el sistema.	
Excepciones	Paso	Acción
	4	Si el sistema <i>no puede guardar la onda, informa al Usuario, a continuación el caso de uso queda sin efecto.</i>

Tabla 39- Descripción de análisis del caso de uso 005



UC-006: Extraer Características

UC-0006	ExtraerCaracterísticas	
Versión	1.0 (13/10/2015)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desee ExtraerCaracterísticas</i>	
Precondición	Que haya una onda cargada en el sistema	
Secuencia normal	Paso	Acción
	1	El actor <u>Usuario(ACT-0001)</u> selecciona <i>Extraer Características</i>
	2	El sistema <i>solicita la información del usuario de la onda a extraer las características de la lista disponible</i>
	3	El actor <u>Usuario(ACT-0001)</u> selecciona <i>la onda a extraer las características</i>
	4	El sistema <i>extrae las características acorde con los requisitos.</i>
	5	El sistema <i>muestra al usuario las características extraídas.</i>
Postcondición	Ninguna	
Excepciones	Paso	Acción
	4	<i>Si el sistema no puede extraer las características de la onda, se notifica al usuario. A continuación el caso de uso queda sin efecto.</i>

Tabla 40- Descripción de análisis del caso de uso 006

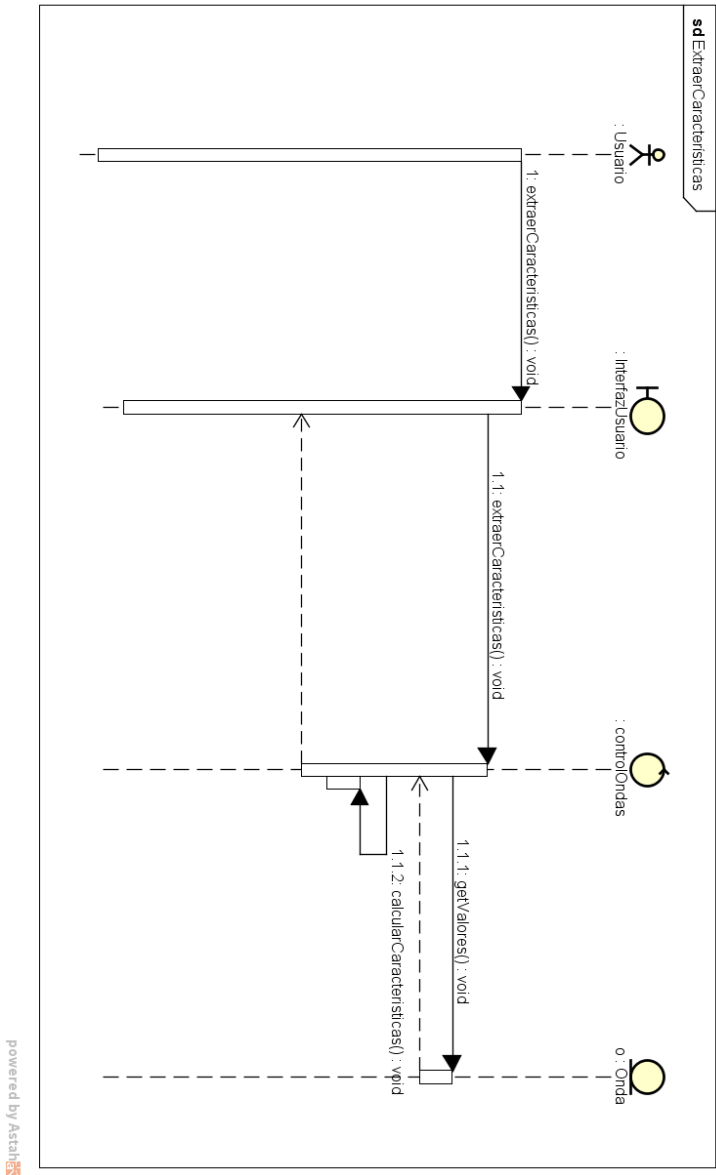
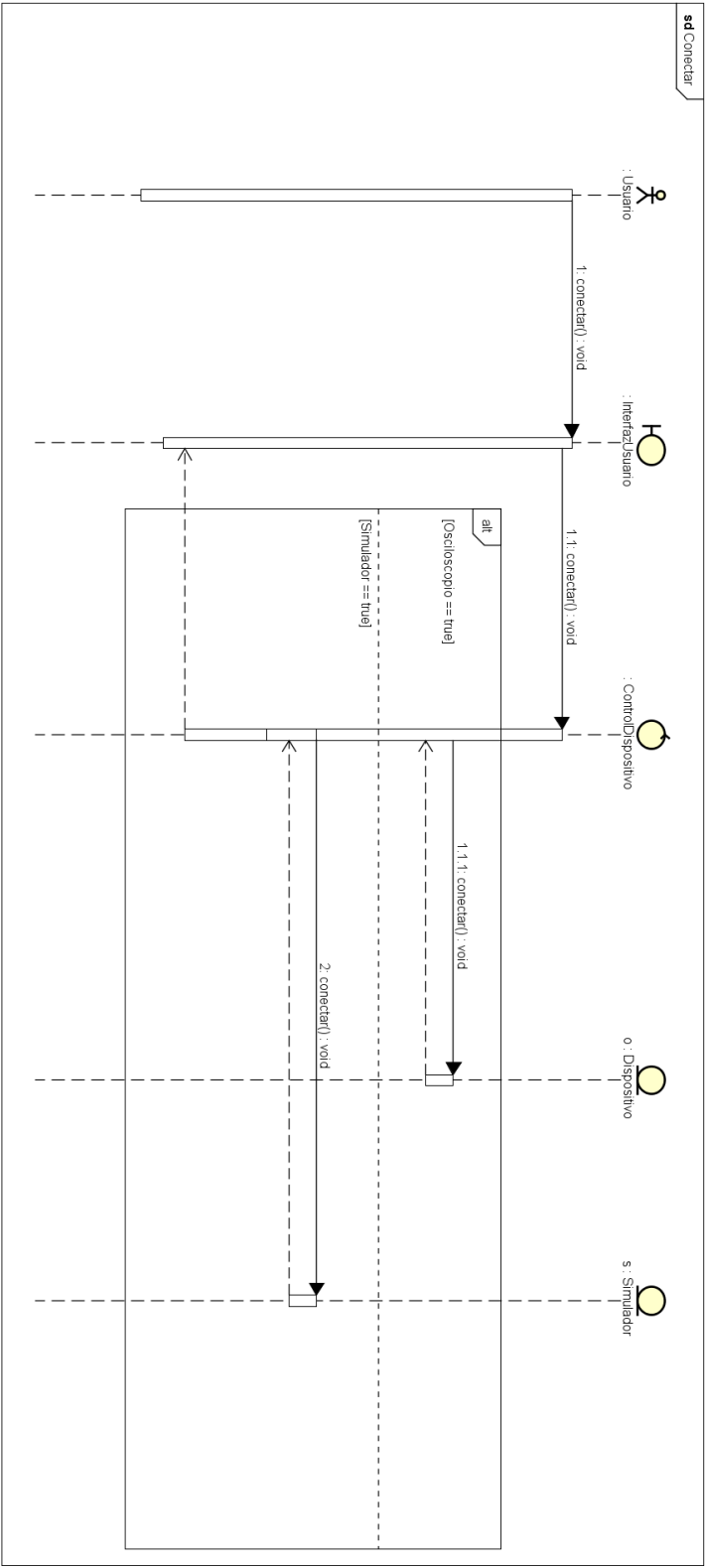


Ilustración 26- Diagrama de secuencia de análisis del caso de uso 006

UC-007: Conectar

UC-0007	Conectar	
Versión	1.0 (13/10/2015)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desee conectarse al osciloscopio o al simulador</i>	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El actor <u>Usuario(ACT-0001)</u> <i>selecciona Conectar</i>
	2	El sistema <i>solicita la información del usuario de conectar a un osciloscopio o bien a un simulador</i>
	3	El actor <u>Usuario(ACT-0001)</u> <i>selecciona la clase de dispositivo al que conectar</i>
	4	El sistema <i>intenta conectarse a la clase de dispositivo seleccionado.</i>
	5	El sistema <i>informa al Usuario del resultado de la conexión.</i>
Postcondición	El sistema queda conectado a un dispositivo	
Excepciones	Paso	Acción
	4	El sistema <i>no puede conectarse al dispositivo, entonces se conecta al simulador, a continuación el caso de uso continua normalmente.</i>

Tabla 41- Descripción de análisis del caso de uso 007



UC-008: Desconectar

UC-0008	Desconectar	
Versión	1.0 (13/10/2015)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desee desconectarse del osciloscopio/simulador</i>	
Precondición	Que el sistema esté conectado a un osciloscopio al simulador	
Secuencia normal	Paso	Acción
	1	El actor <u>Usuario(ACT-0001)</u> <i>selecciona Desconectar</i>
	2	El sistema <i>solicita la desconexión del osciloscopio/simulador</i>
	3	El sistema <i>informa del resultado al Usuario</i>
Postcondición	El sistema queda desconectado del osciloscopio/simulador	
Excepciones	Paso	Acción

Tabla 42- Descripción de análisis del caso de uso 008

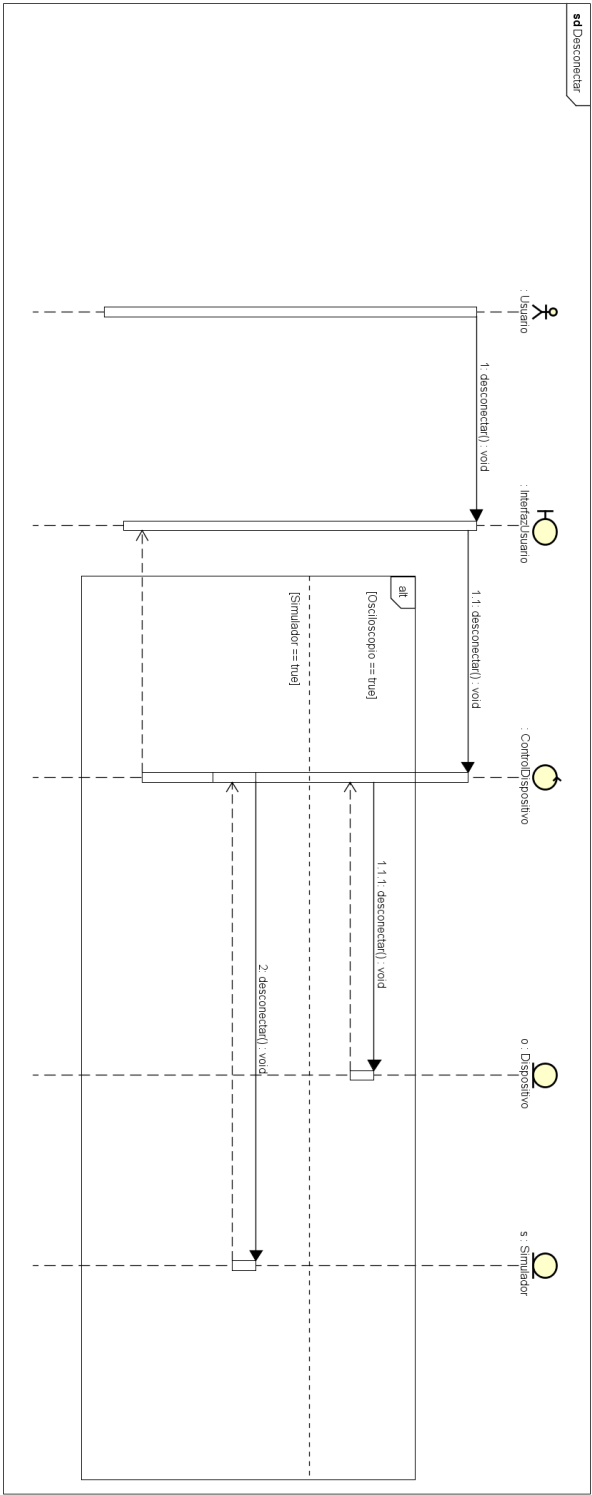
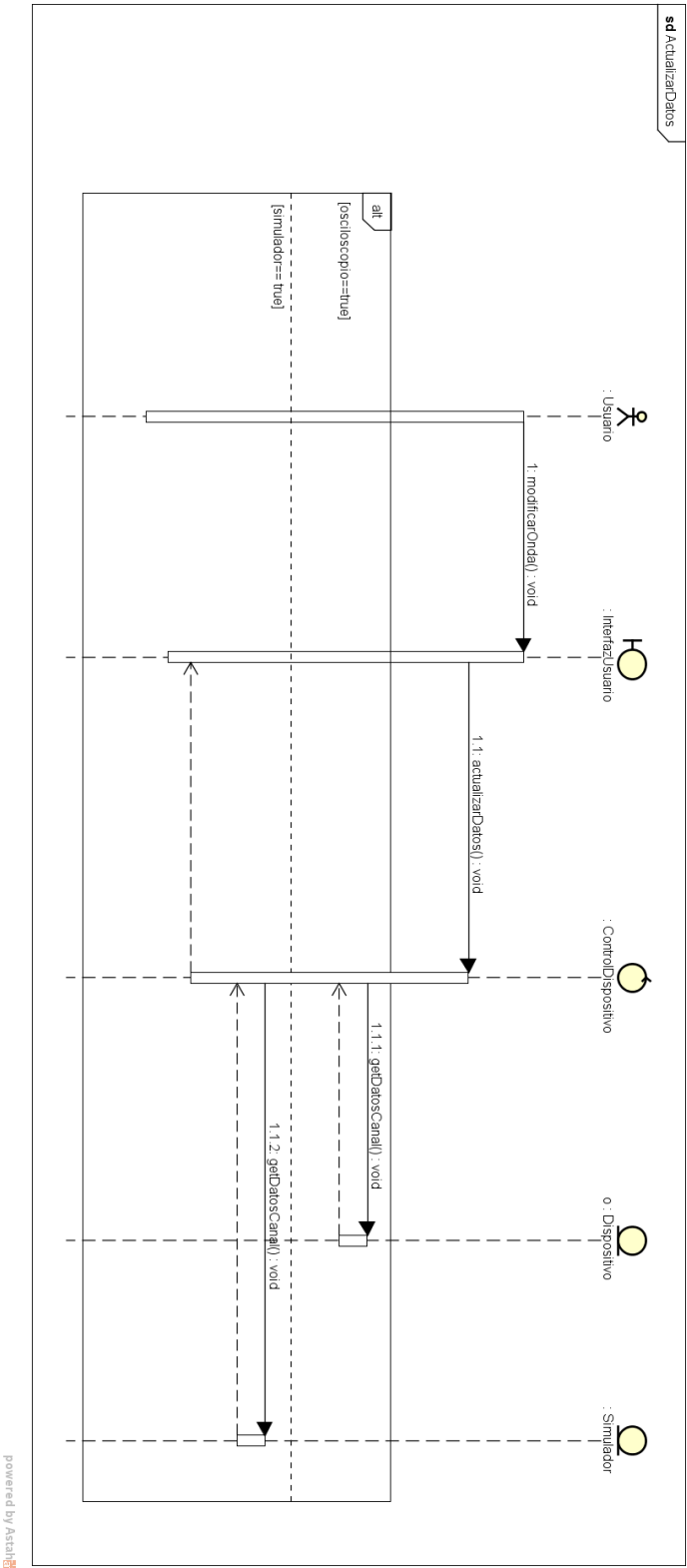


Ilustración 28- Diagrama de secuencia de análisis del caso de uso 008

UC-009: Actualizar Datos

UC-0009	ActualizarDatos	
Versión	1.0 (13/10/2015)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el Usuario desee actualizar datos</i>	
Precondición	Que el sistema esté conectado con el simulador, o osciloscopio	
Secuencia normal	Paso	Acción
	1	El actor <u>Usuario(ACT-0001)</u> <i>selecciona ActualizarDatos</i>
	2	El sistema <i>comprueba el canal para el cual se van a actualizar los datos.</i>
	3	El sistema <i>solicita los datos del canal al osciloscopio o al simulador.</i>
	4	El sistema <i>muestra los datos recibidos del canal al usuario.</i>
Postcondición		
Excepciones	Paso	Acción
	3	<i>Si el sistema obtiene una respuesta negativa del osciloscopio o simulador, se informa al Usuario, a continuación el caso de uso queda sin efecto.</i>

Tabla 43- Descripción de análisis del caso de uso 009



UC-010: Cambiar Canal

UC-0010	CambiarCanal	
Versión	1.0 (13/10/2015)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desee cambiar el canal del osciloscopio/simulador</i>	
Precondición	Que la aplicación esté conectado al osciloscopio / simulador	
Secuencia normal	Paso	Acción
	1	El actor Usuario(ACT-0001) <i>selecciona Cambiar Canal</i>
	2	El sistema <i>solicita la información al osciloscopio/simulador de los canales disponibles</i>
	3	El sistema <i>muestra la lista de canales disponibles al Usuario</i>
	4	El actor Usuario(ACT-0001) <i>selecciona el canal a cambiar</i>
	5	El sistema <i>solicita la conexión al canal seleccionado al osciloscopio/simulador</i>
	6	El sistema <i>ha cambiado de canal.</i>
Postcondición	El sistema está conectado al nuevo canal	
Excepciones	Paso	Acción
	5	El sistema <i>no puede conectarse al nuevo canal, se invalida el cambio de canal y a continuación el caso de uso queda sin efecto.</i>

Tabla 44- Descripción de análisis del caso de uso 010

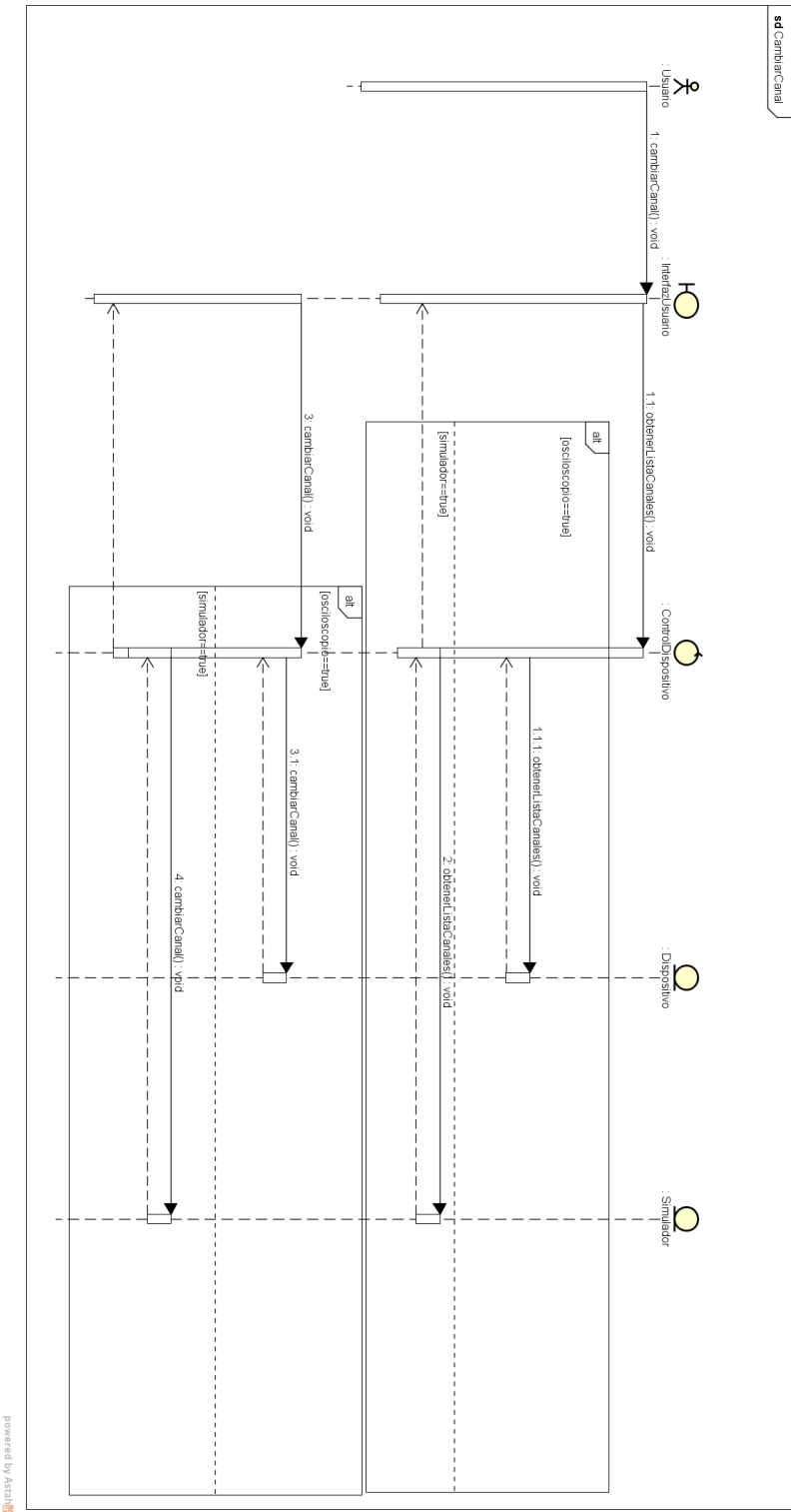
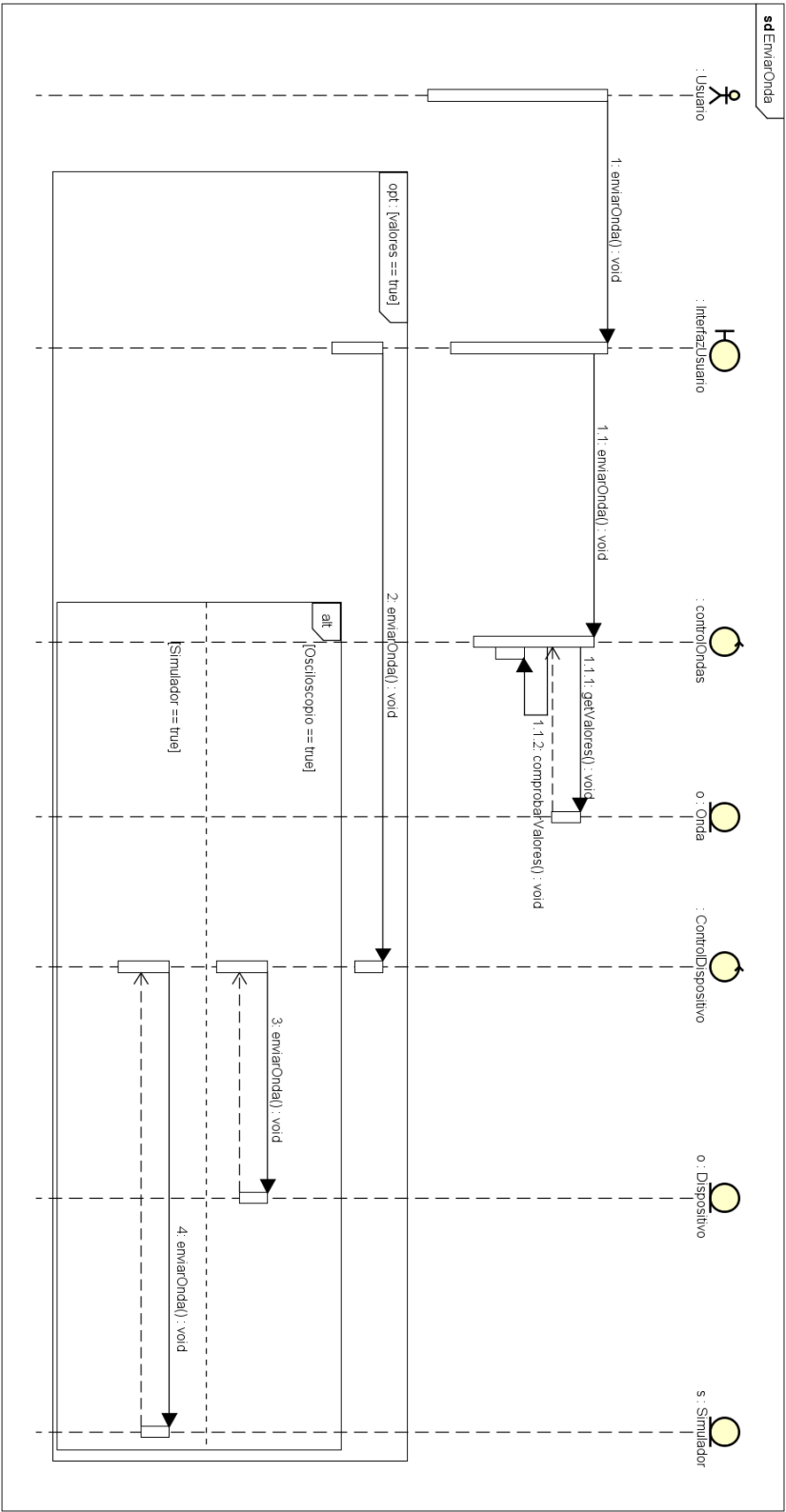


Ilustración 30-Diagrama de secuencia de análisis del caso de uso 010

UC-011: Enviar Onda

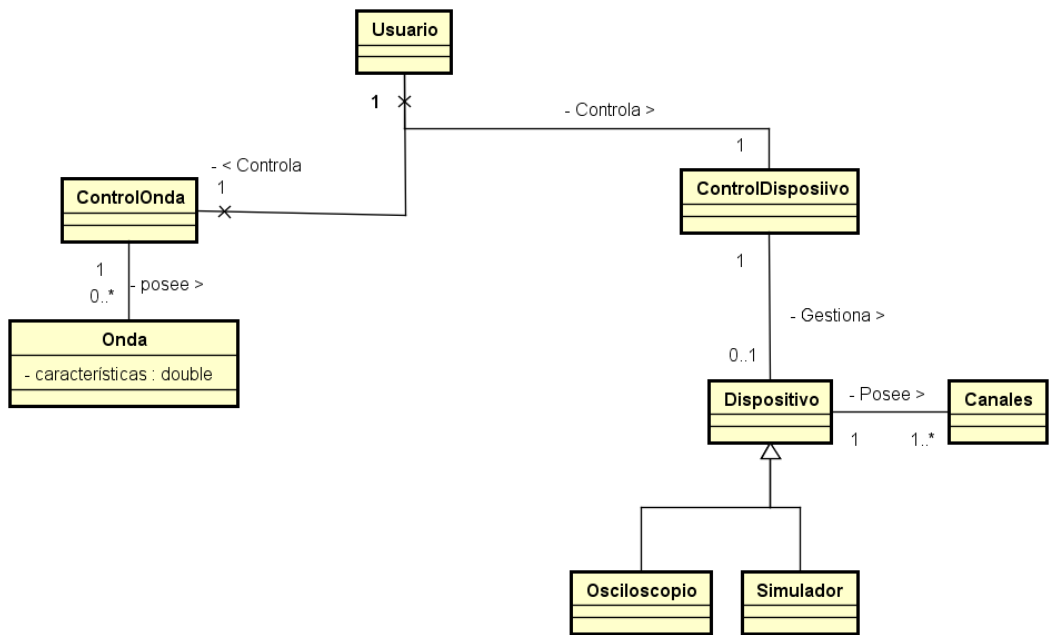
UC-0011	EnviarOnda	
Versión	1.0 (13/10/2015)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desee enviar una onda.</i>	
Precondición	Que la aplicación esté conectado al osciloscopio / simulador. Que haya una onda cargada en el sistema.	
Secuencia normal	Paso	Acción
	1	El actor <u>Usuario(ACT-0001)</u> selecciona <i>Enviar Onda.</i>
	2	El sistema <i>lee la onda cargada previamente.</i>
	3	El sistema <i>envía la onda solicitada al dispositivo.</i>
Postcondición	El sistema ha enviado la onda	
Excepciones	Paso	Acción
	2	El sistema <i>no puede leer la onda cargada correctamente, se invalida el envío de onda y a continuación el caso de uso queda sin efecto.</i>

Tabla 45- Descripción de análisis del caso de uso 011



4.5. Modelo de Dominio

Se incluye a continuación el diagrama correspondiente al modelo de dominio.



powered by Astah

Ilustración 32- Diagrama del modelo de dominio

4.6. Diagrama de clases de análisis

Según Arlow [Arl06] y el proceso de software asociado, OpenUP, se establece en el análisis un modelo para encontrar las clases correspondientes a esta etapa. Este modelo es llamado por sus siglas BCE, es decir, Boundary-Control-Entity.

Según este modelo, las clases de análisis se pueden dividir en tres categorías que explicamos a continuación:

Boundary: Son las clases que existen en el límite del sistema y se comunican con los actores externos.

Control: Estas clases, son controladores, sus instancias coordinan comportamiento del sistema que corresponde a uno o más casos de uso.

Entity: Estas clases modelan información sobre algo y normalmente tienen un comportamiento muy sencillo que equivale a poco más que a obtener y establecer valores. Normalmente suele representar información persistente.

La razón, de porqué se ha establecido este modelo a la hora de identificar las clases del análisis, es porque en el diseño va a facilitar el traslado a la arquitectura propuesta por el FrameWork, que es MVVM.

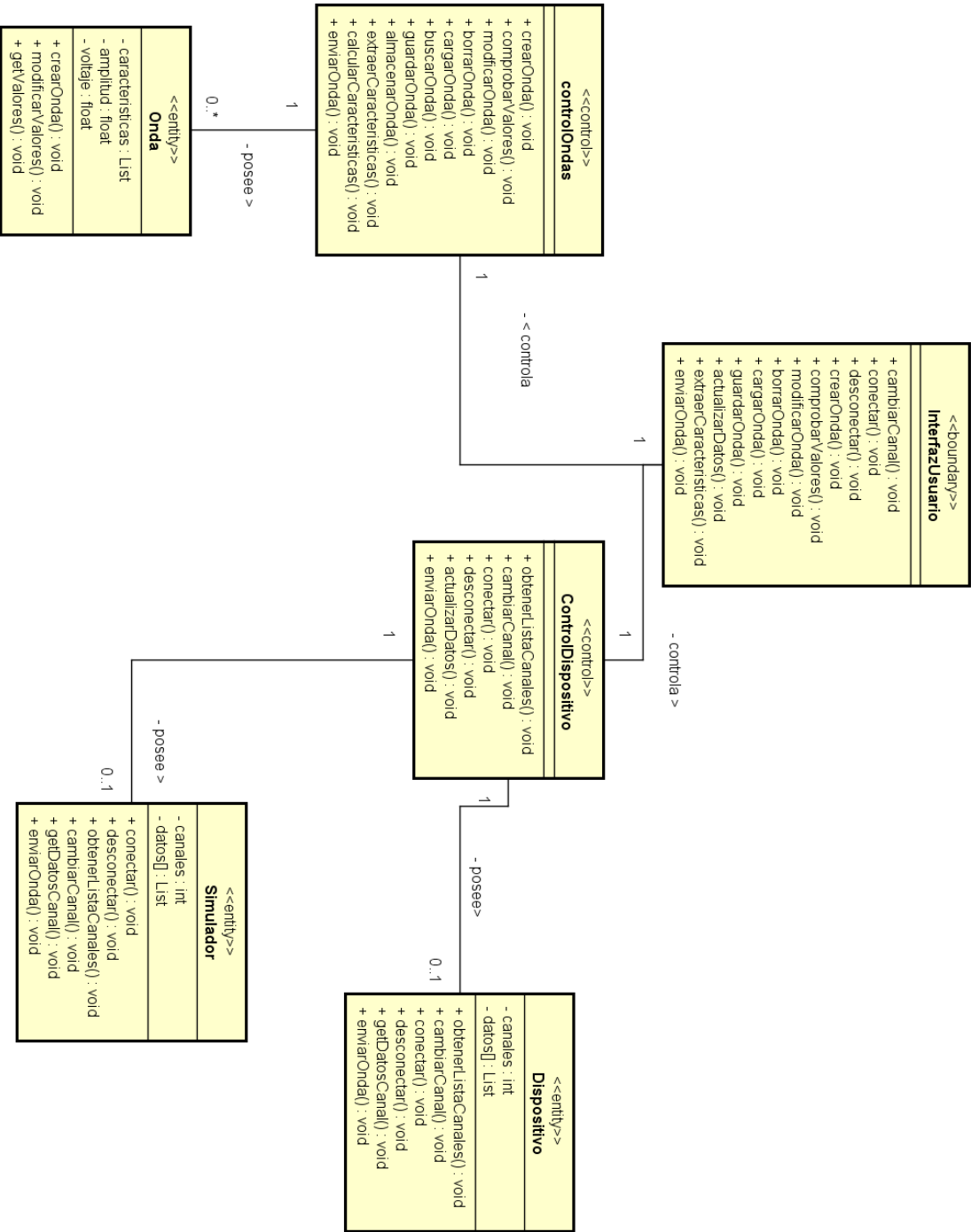


Ilustración 33- Diagrama de clases de análisis

5. Capítulo quinto: DISEÑO DE LA APLICACIÓN

El diseño de software, según Sommerville [38] [47], describe la estructura del software que se va a implementar, los datos que forman parte del sistema, así como las interfaces entre los distintos componentes y, algunas veces, los algoritmos utilizados.

En este capítulo, y debido al proceso seguido (OpenUP) se describirán los siguientes artefactos:

- Documento de la arquitectura física y lógica del sistema:
- Documento de casos de uso de diseño
- Documento de clases de diseño

5.1. Objetivos del diseño

El diseño especificado en este documento se ha elaborado buscando un sistema capaz de cumplir los objetivos especificados en el documento de análisis.

Eficiencia: Se buscará la eficiencia de la aplicación, a fin de no retrasar la actualización de los datos recibidas en la misma.

Usabilidad: El sistema a desarrollar será sencillo de utilizar para el cliente, presentando un estudio de la experiencia del usuario. El diseño de la interfaz se puede encontrar en este capítulo.

Corrección: Se buscará un alto grado de corrección, comprobando que los datos obtenidos por la aplicación contienen una tasa baja de fallos.

5.2. Arquitectura Lógica

Para la arquitectura lógica del sistema, al utilizar como marco de desarrollo .NET Framework y WPF (Windows Presentation Foundation) [48] la aplicación queda restringida al patrón MVP (Modelo-Vista- Presentador) [49], patrón arquitectónico, el cual se compone de los siguientes elementos:

- **Vista:** La vista es responsable de definir la estructura y apariencia de lo que el usuario ve en pantalla. En este patrón, se elimina el comportamiento o lógica de negocio que lo conforma lo que permite utilizar un meta-lenguaje para su definición, en este caso XAML. La iteración entre esta capa y la lógica de negocio se puede realizar utilizando “bindings” entre los datos y la capa de modelo, siguiendo el modelo MVVM puro, o bien dividiendo la logica de negocio entre la capa de modelo y la clase en C# correspondiente a la vista (clase vista-modelo), similar a seguir un patrón MVC (Modelo-Vista-Controlador) con un controlador débil.
- **Modelo:** El modelo en MVVM es una implementación del dominio de la aplicación que incluye la lógica de negocio y validación.
- **Presentación:** La clase de presentación actúa como un intermediario entre la vista y el modelo y es el responsable de manejar la lógica de la vista. Típicamente, la clase de presentación interactúa con el modelo invocando las clases de este. También transforma los datos recibidos del modelo de una manera que la vista pueda interpretar. La clase de presentación proporciona además implementaciones de los comandos que un usuario de la aplicación inicia en la vista, así como el control de la variación del estado del aspecto de la vista.

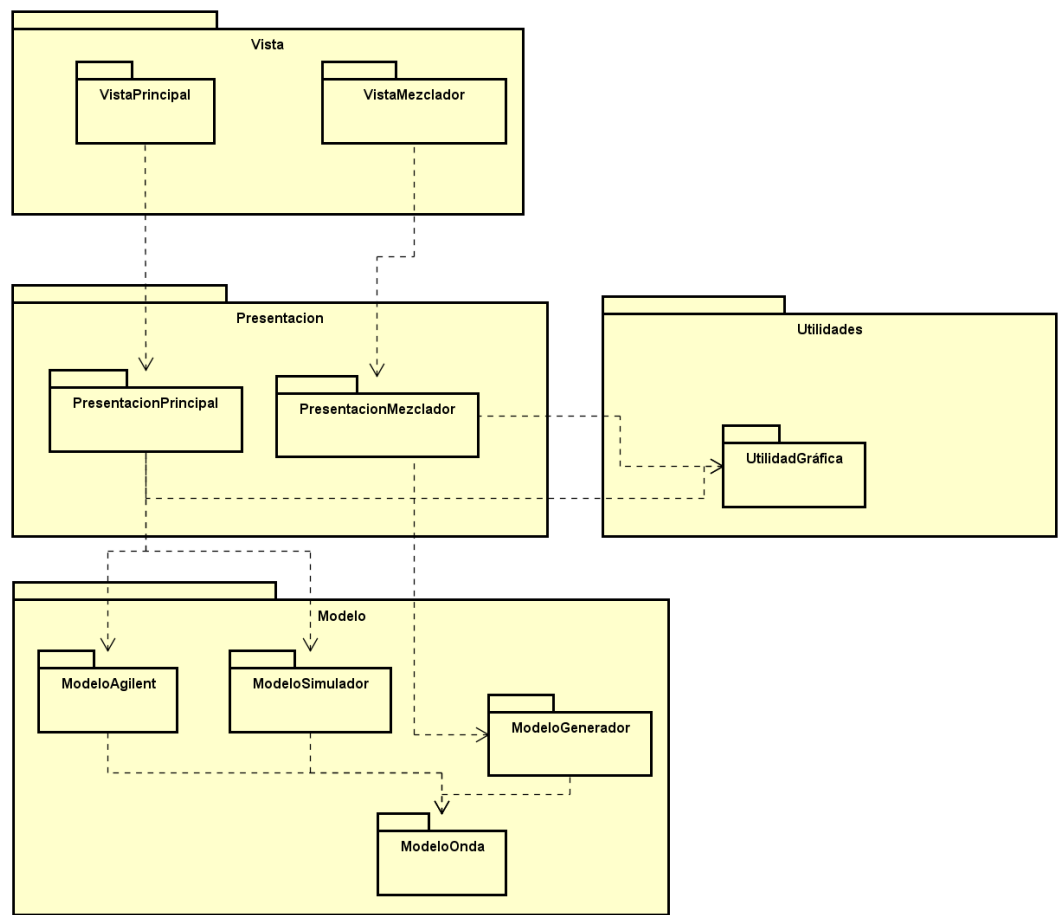
Elección de la arquitectura

Aunque se ha citado anteriormente, la elección de la arquitectura viene marcada por el framework de desarrollo elegido, en este caso, .NET Framework y WPF, cabe citar que en caso de haber sido desarrollada la aplicación en Silverlight, el framework .NET obliga a utilizar en su lugar MVC debido al carácter distribuido de esta.

No obstante, algunas de las ventajas de usar MVP sobre MVC [50] son que se reduce el código necesario para realizar la aplicación y se logra en función, de la versión de MVP utilizada una separación alta entre la vista y el modelo.

Diagrama de la arquitectura lógica del sistema

El diagrama correspondiente a la arquitectura lógica de la aplicación es el siguiente:



powered by Astah

Ilustración 34- Diagrama de la arquitectura lógica

El paquete de utilidades, se desarrolló con la siguiente finalidad:
Separar en componente la actualización de la pantalla con la finalidad de mejorar la eficiencia a la hora de dibujar la gráfica con los datos recibidos del osciloscopio.

5.3. Arquitectura física del sistema

El dispositivo de instalación es un ordenador con el sistema operativo Windows 8.1 y el framework .NET instalados. La aplicación presentada utilizará las bibliotecas correspondientes al marco de desarrollo citado anteriormente, y de los drivers de Keysight para controlar el osciloscopio a través del puerto USB. En caso de conectar el amplificador lock-in se utilizará la interfaz Keysight 82357B, para enviar comandos vía protocolo GPIB por el puerto USB.

El diagrama correspondiente a la arquitectura física es extraído de [1] y es el siguiente:

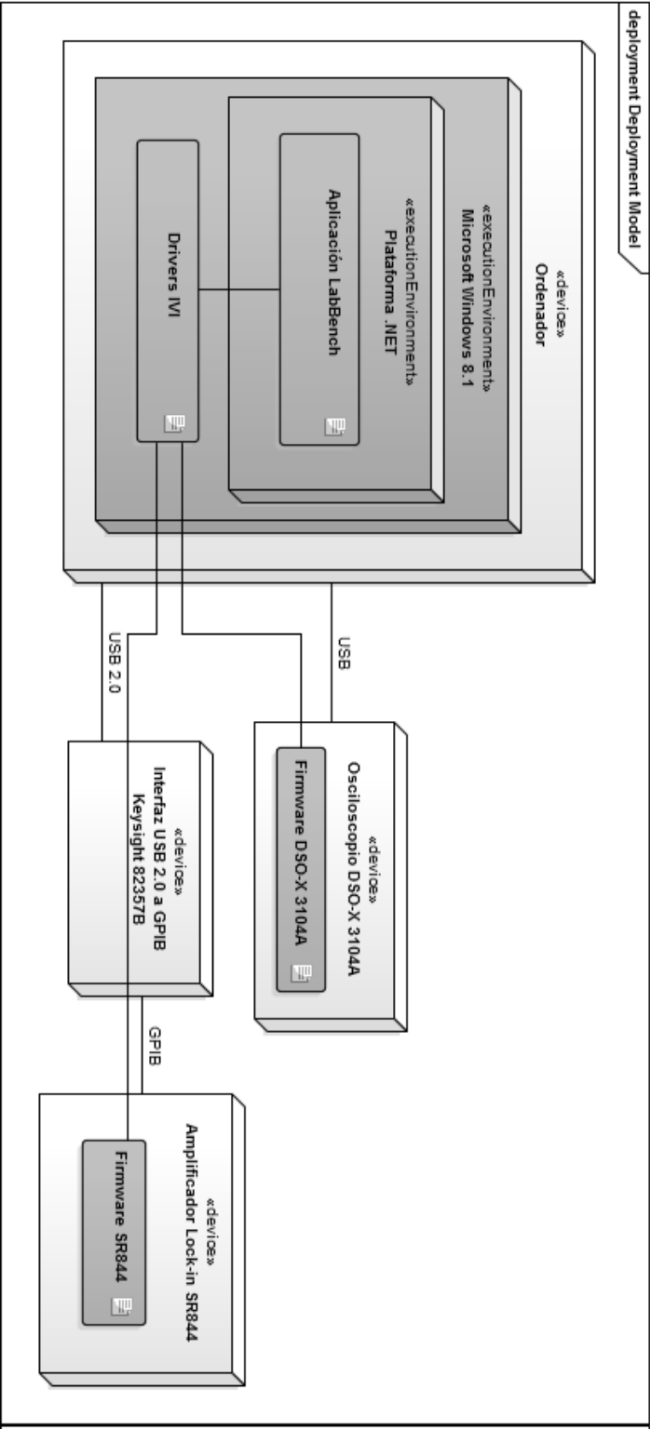


Ilustración 35- Diagrama de la arquitectura física

5.4.Descripción de los casos de uso de diseño

Al igual que en análisis, en diseño se incluye por cada diagrama, su diagrama de secuencia correspondiente.

UC-001: Crear Onda

UC-0001	Crear Onda	
Versión	1.0 (13/10/2015)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desee crear una onda</i>	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El actor <u>Usuario(ACT-0001)</u> selecciona <i>Crear Onda</i>
	2	El sistema <i>solicita la información del usuario de la onda a crear</i>
	3	El actor <u>Usuario(ACT-0001)</u> introduce la <i>información de la onda a crear</i>
	4	El sistema <i>comprueba los valores introducidos</i>
	5	El sistema <i>calcula la suma de senos correspondiente.</i>
	6	El sistema <i>almacena los datos de la onda creada.</i>
	7	El sistema <i>dibuja y muestra la onda creada al usuario.</i>
Postcondición	La nueva onda queda almacenada en el sistema	
Excepciones	Paso	Acción
	4	Si el Actor <i>Usuario</i> , ha introducido una <i>información errónea</i> , el sistema <i>invalida los datos introducidos y a continuación el caso de uso queda sin efecto.</i>

Tabla 46 - Descripción de diseño del caso de uso 001

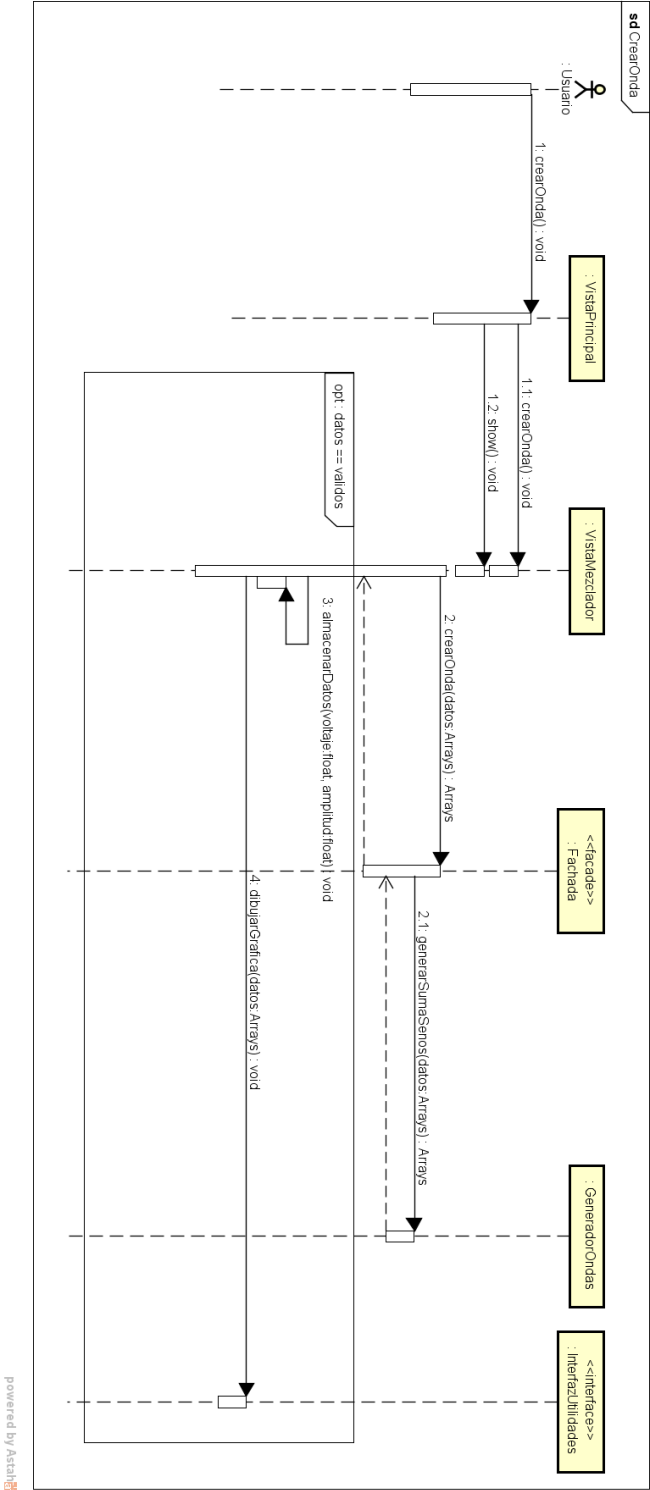
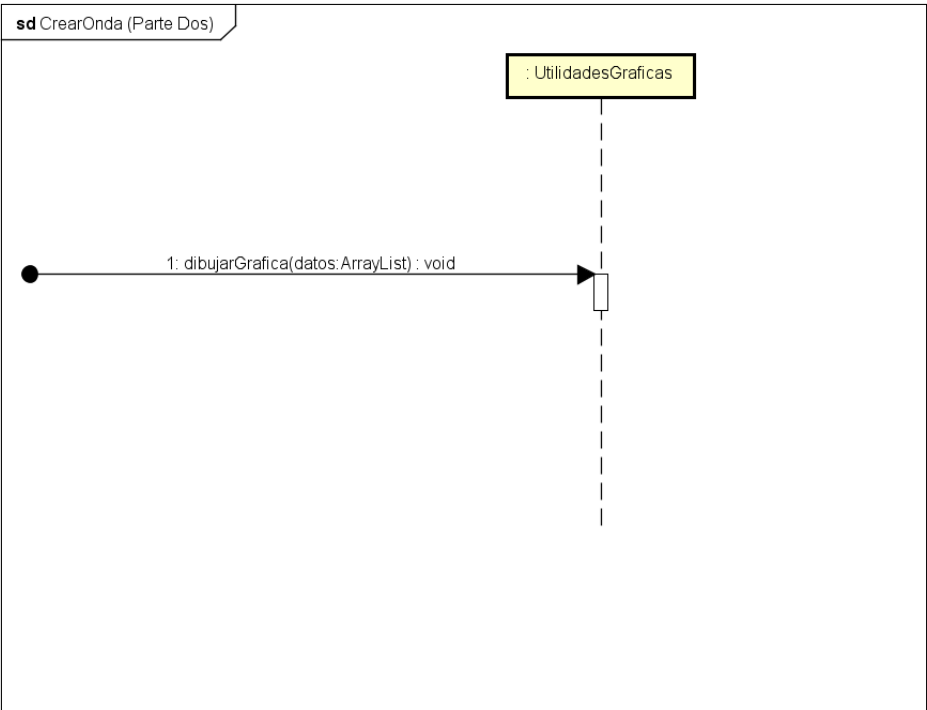


Ilustración 36- Diagrama de secuencia de diseño del caso de uso 001



powered by Astah

Ilustración 37- Diagrama de secuencia de diseño del caso de uso 001 (Parte dos)

UC-002: Modificar Onda

UC-0002	Modificar Onda	
Versión	1.0 (13/10/2015)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desee modificar una onda</i>	
Precondición	Que haya una onda cargada en el sistema	
Secuencia normal	Paso	Acción
	1	El actor <u>Usuario(ACT-0001)</u> selecciona <i>Modificar Onda</i>
	2	El sistema <i>solicita la información del usuario de la onda a modificar de la lista disponible</i>
	3	El actor <u>Usuario(ACT-0001)</u> selecciona <i>la onda a modificar</i>
	4	El sistema <i>solicita la información a modificar de la onda seleccionada</i>
	5	El actor <u>Usuario(ACT-0001)</u> introduce la nueva <i>información de la onda a modificar</i>
	6	El sistema <i>valida la información introducida.</i>
	7	El sistema <i>calcula la suma de senos correspondiente a la nueva onda.</i>
	8	El sistema <i>almacena los datos de la nueva onda.</i>
	9	El sistema <i>dibuja y muestra la onda al usuario.</i>
Postcondición	La onda modificada queda almacenada en el sistema	
Excepciones	Paso	Acción
	6	Si el Actor Usuario, ha introducido una <i>información errónea, el sistema invalida los datos introducidos y a continuación el caso de uso queda sin efecto.</i>

Tabla 47- Descripción de diseño del caso de uso 002

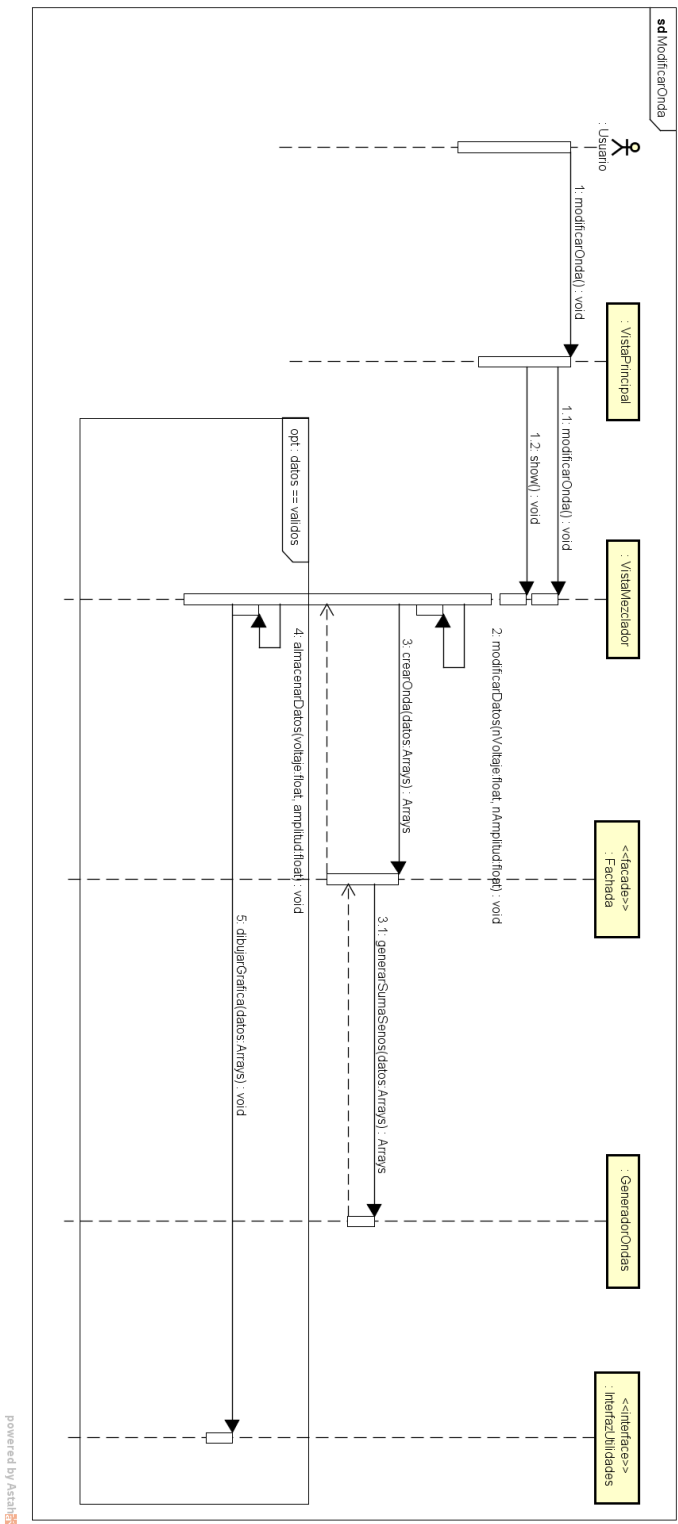
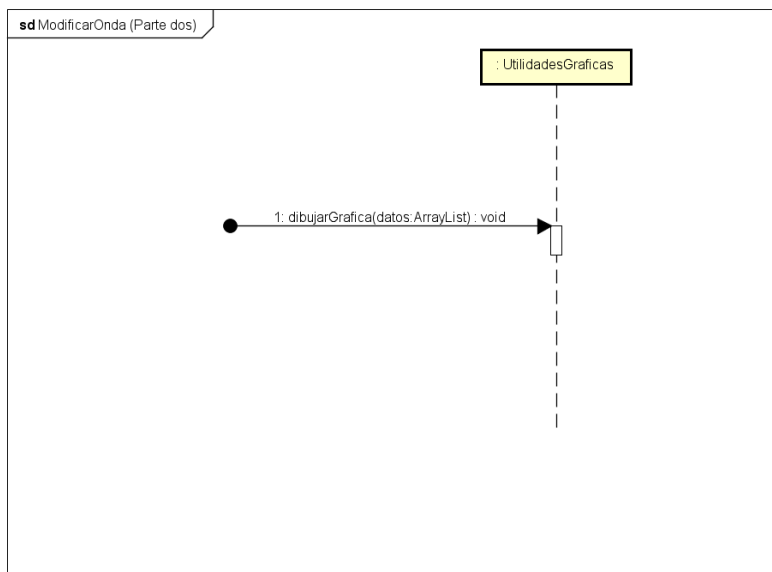


Ilustración 38- Diagrama de secuencia de diseño del caso de uso 002



powered by Astah

Ilustración 39- Diagrama de secuencia de diseño del caso de uso 002 (Parte dos)

UC-003: Borrar Onda

UC-0003	BorrarOnda	
Versión	1.0 (13/10/2015)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desee borrar una onda</i>	
Precondición	Que haya una onda cargada en el sistema	
Secuencia normal	Paso	Acción
	1	El actor <u>Usuario(ACT-0001)</u> <i>selecciona BorrarOnda</i>
	2	El sistema <i>solicita la información del usuario de la onda a eliminar de la lista disponible</i>
	3	El actor <u>Usuario(ACT-0001)</u> <i>selecciona la onda a eliminar</i>
	4	El sistema <i>elimina la onda seleccionada.</i>
	5	El sistema <i>recalcula para la onda restante la suma de senos</i>
	6	El sistema <i>almacena los datos de la onda sin la componente eliminada.</i>
	7	El sistema <i>dibuja y muestra la onda al usuario.</i>
Postcondición	La onda queda eliminada del sistema.	
Excepciones	Paso	Acción
	3	<i>Si la onda eliminada, era una onda simple el caso de uso finaliza en este punto.</i>

Tabla 48- Descripción de diseño del caso de uso 003

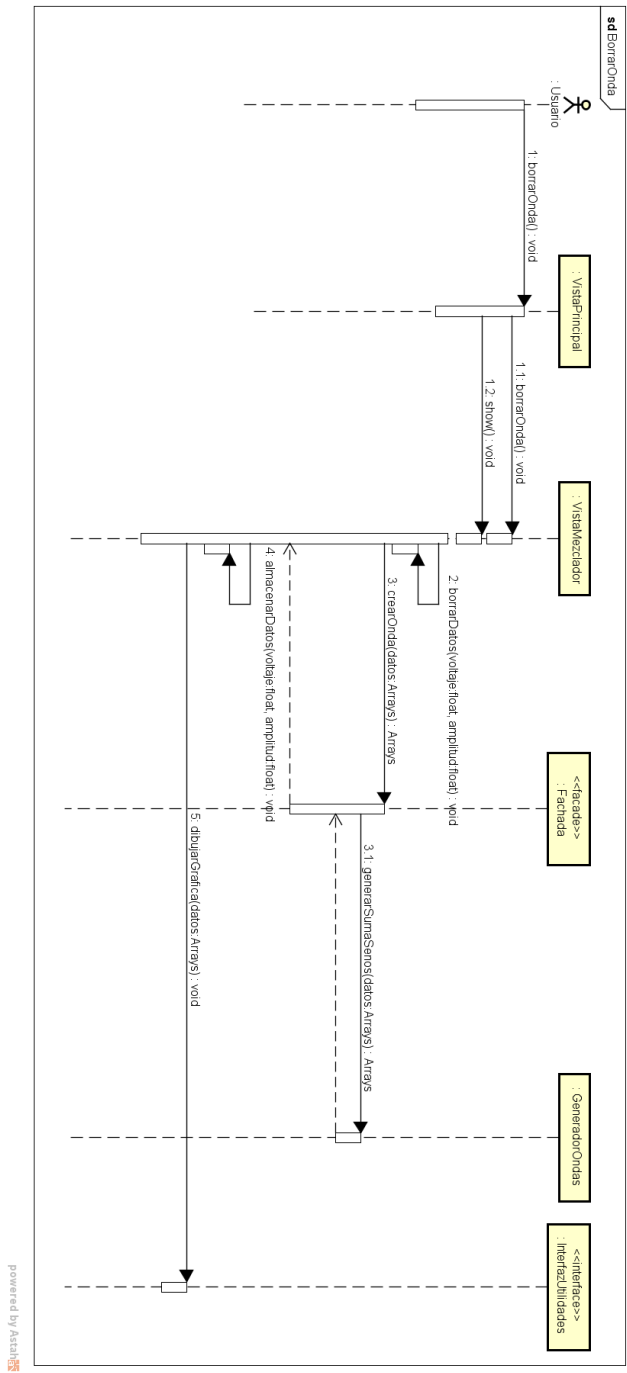


Ilustración 40- Diagrama de secuencia de diseño del caso de uso 003

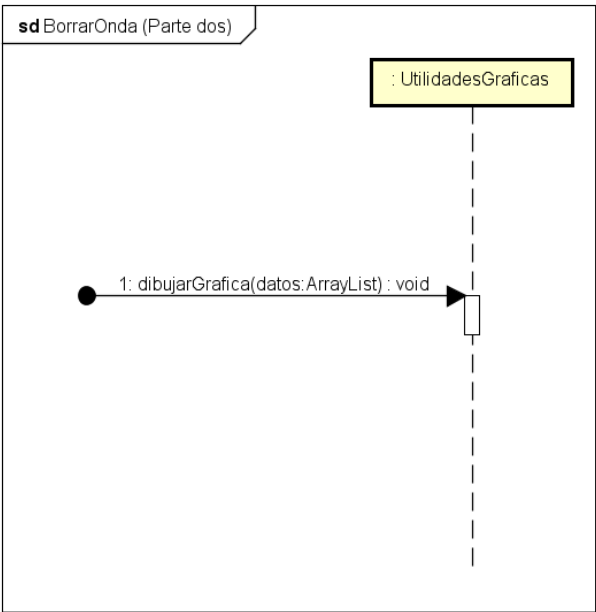


Ilustración 41- Diagrama de secuencia de diseño del caso de uso 003 (parte dos)

UC-004: Cargar Onda

UC-0004	CargarOnda	
Versión	1.0 (13/10/2015)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desee cargar una onda</i>	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El actor Usuario(ACT-0001) <i>selecciona CargarOnda</i>
	2	El sistema <i>solicita la información del usuario de la onda a cargar</i>
	3	El actor Usuario(ACT-0001) <i>introduce la ruta de la onda a cargar</i>
	4	El sistema <i>calcula la suma de senos para la onda cargada.</i>
	5	El sistema <i>dibuja y muestra la onda cargada</i>
Postcondición	La onda queda cargada en el sistema	
Excepciones	Paso	Acción
	4	Si el sistema <i>no es capaz de cargar la onda correctamente, notifica al Usuario, a continuación el caso de uso queda sin efecto</i>

Tabla 49- Descripción de diseño del caso de uso 004

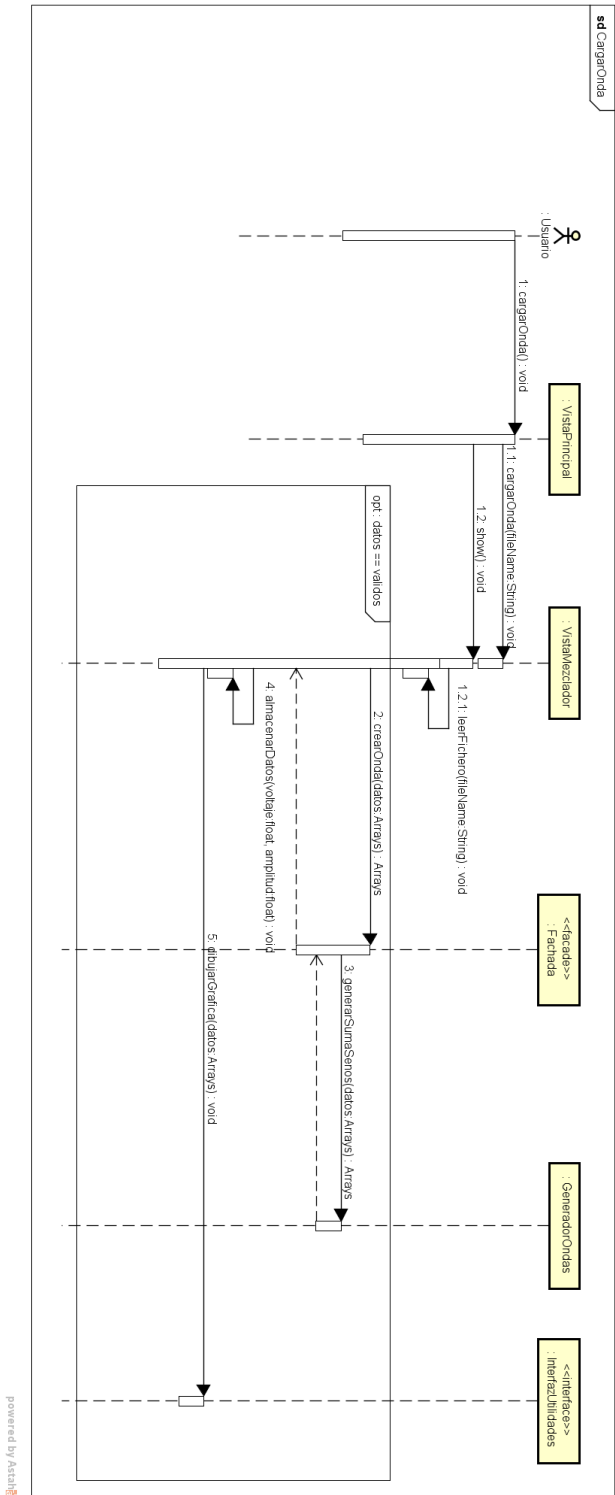


Ilustración 42- Diagrama de secuencia de diseño del caso de uso 004

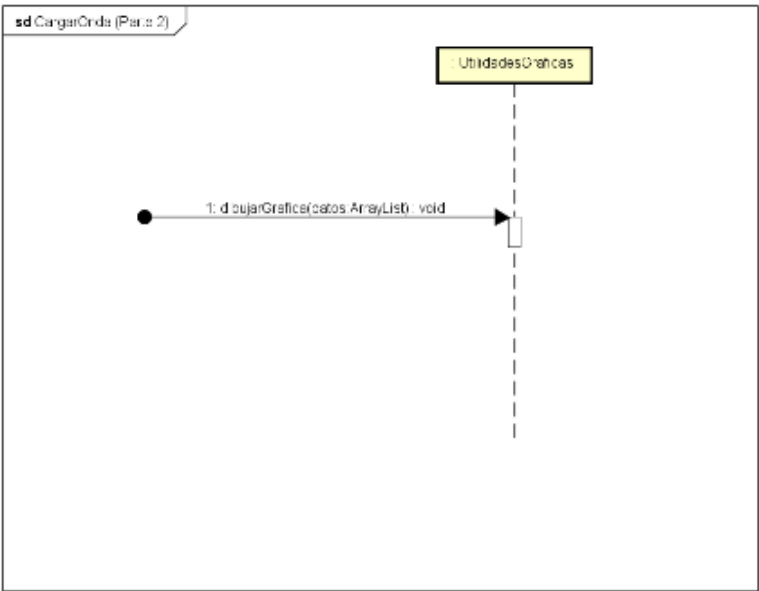


Ilustración 43- Diagrama de secuencia de diseño del caso de uso 004 (Parte dos)

UC-005: Guardar Onda

UC-0005	GuardarOnda	
Versión	1.0 (13/10/2015)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desee guardar una onda</i>	
Precondición	Que haya una onda cargada en el sistema	
Secuencia normal	Paso	Acción
	1	El actor <u>Usuario(ACT-0001)</u> <i>selecciona GuardarOnda</i>
	2	El sistema <i>solicita la información del usuario de la onda a guardar de la lista disponible</i>
	3	El actor <u>Usuario(ACT-0001)</u> <i>selecciona la onda a guardar</i>
	4	El sistema <i>almacena los pares de amplitud y voltaje de la onda a guardar.</i>
Postcondición	La onda queda guardada en el sistema.	
Excepciones	Paso	Acción
	4	Si el sistema <i>no puede guardar la onda, informa al Usuario, a continuación el caso de uso queda sin efecto.</i>

Tabla 50- Descripción de diseño del caso de uso 005

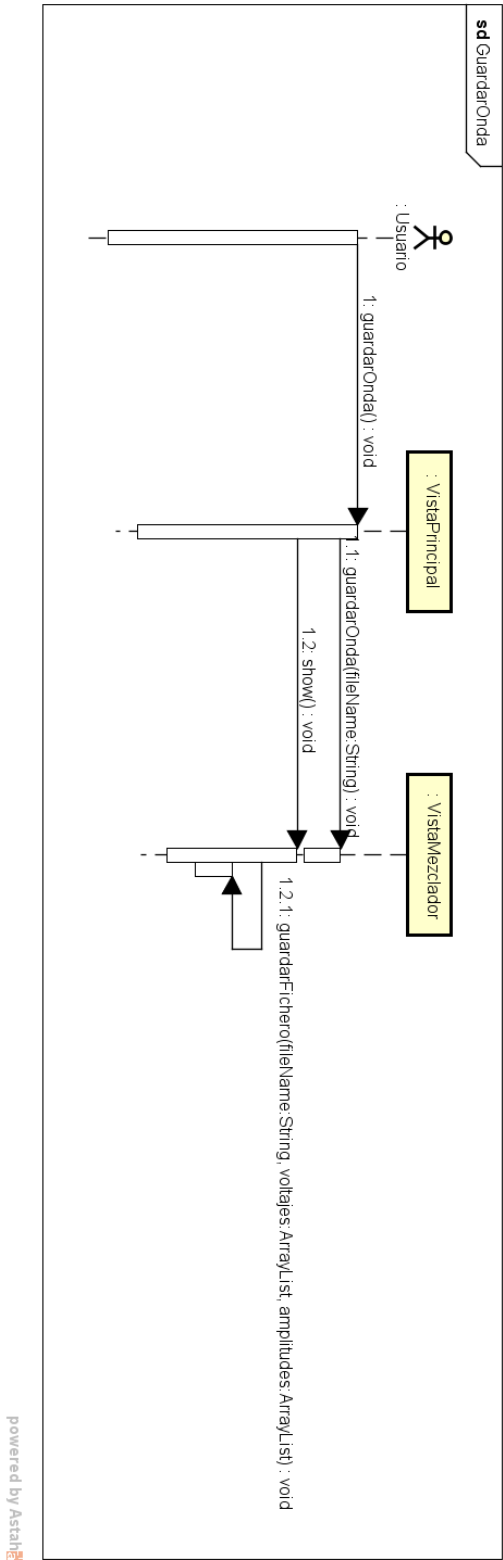


Ilustración 44- Diagrama de secuencia de diseño del caso de uso 005

UC-006: Extraer Características

UC-0006	ExtraerCaracterísticas	
Versión	1.0 (13/10/2015)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario deseé ExtraerCaracterísticas</i>	
Precondición	Que haya una onda cargada en el sistema	
Secuencia normal	Paso	Acción
	1	El actor <u>Usuario(ACT-0001)</u> <i>selecciona Extraer Características</i>
	2	El sistema <i>solicita la información del usuario de la onda a extraer las características de la lista disponible</i>
	3	El actor <u>Usuario(ACT-0001)</u> <i>selecciona la onda a extraer las características</i>
	4	El sistema <i>extrae las características acorde con los requisitos.</i>
	5	El sistema <i>muestra al usuario las características extraídas.</i>
Postcondición	Ninguna	
Excepciones	Paso	Acción
	4	<i>Si el sistema no puede extraer las características de la onda, se notifica al usuario. A continuación el caso de uso queda sin efecto.</i>

Tabla 51- Descripción de diseño del caso de uso 006

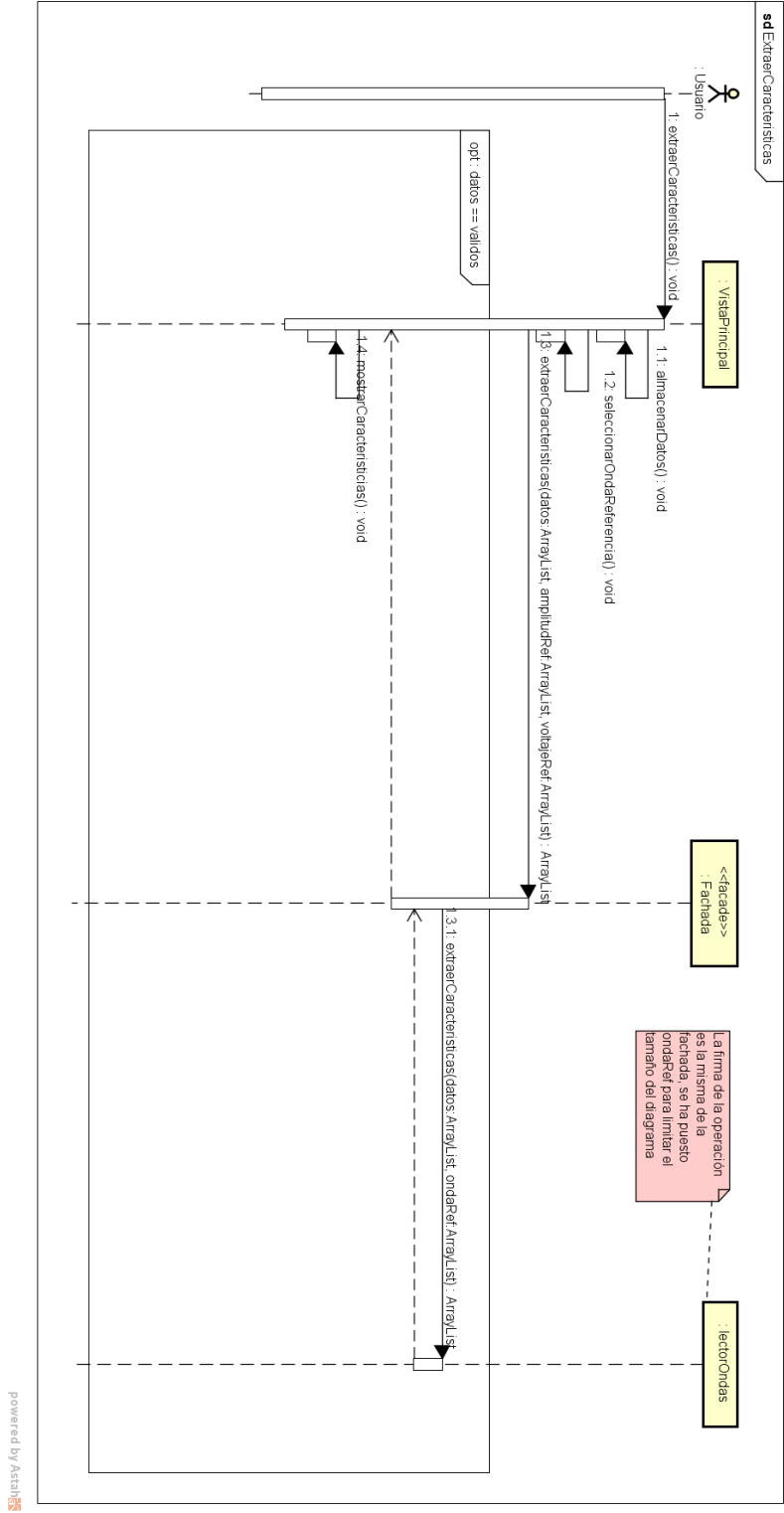


Ilustración 45-Diagrama de secuencia de diseño del caso de uso 006

UC-007: Conectar

UC-0007	Conectar	
Versión	1.0 (13/10/2015)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desee conectarse al osciloscopio o al simulador</i>	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El actor Usuario(ACT-0001) selecciona <i>Conectar</i>
	2	El sistema <i>solicita la información del usuario de conectar a un osciloscopio o bien a un simulador</i>
	3	El actor Usuario(ACT-0001) selecciona <i>la clase de dispositivo al que conectar</i>
	4	El sistema <i>intenta conectarse a la clase de dispositivo seleccionado.</i>
	5	El sistema <i>recupera la lista de canales disponibles del osciloscopio o bien del simulador.</i>
	6	El sistema <i>informa al Usuario del resultado de la conexión.</i>
Postcondición	El sistema queda conectado a un dispositivo	
Excepciones	Paso	Acción
	4	El sistema <i>no puede conectarse al dispositivo, entonces se conecta al simulador, a continuación el caso de uso continua normalmente.</i>

Tabla 52- Descripción de diseño del caso de uso 007

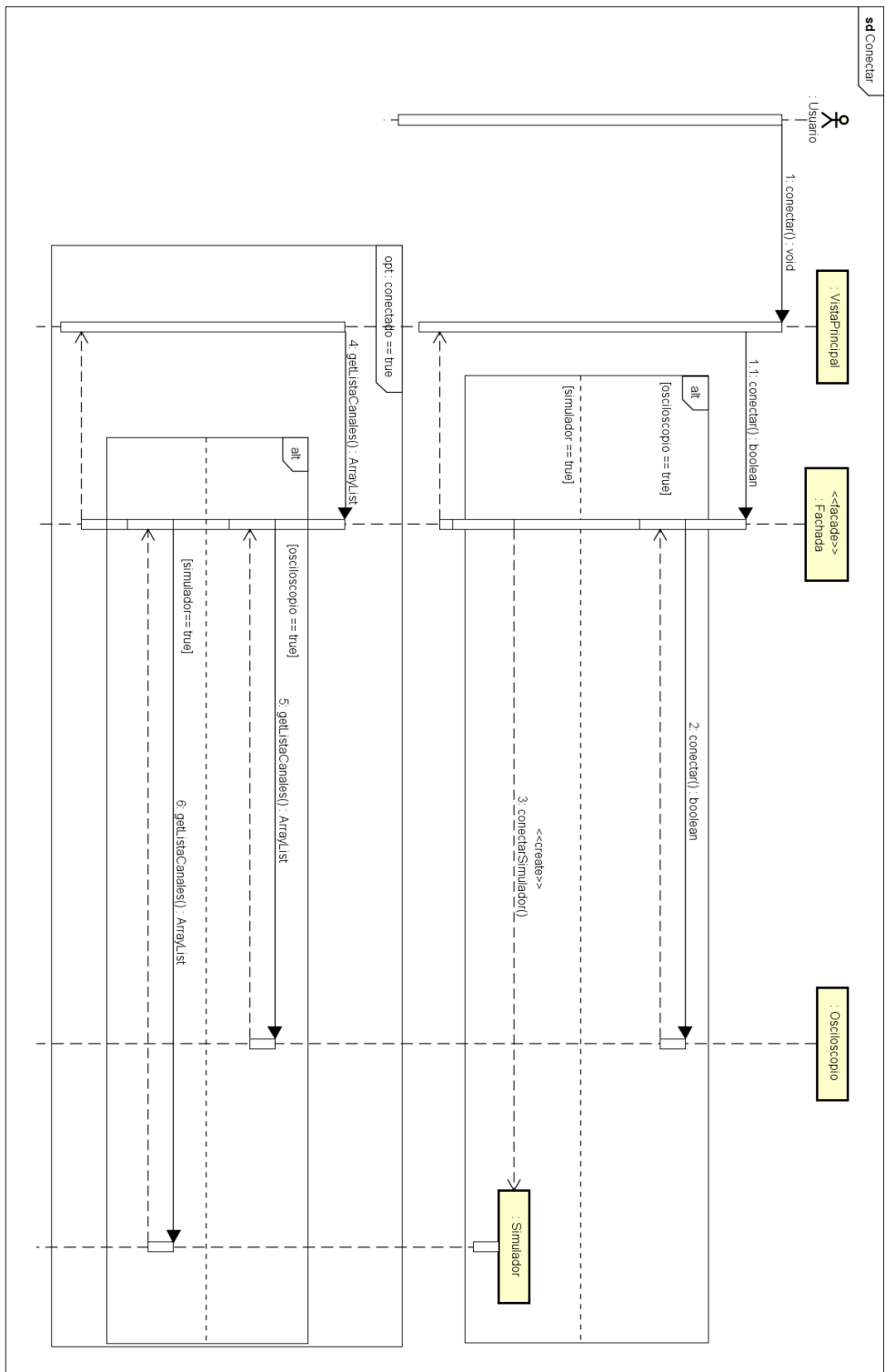


Ilustración 46-Diagrama de secuencia de diseño del caso de uso 007

UC-008: Desconectar

UC-0008	Desconectar	
Versión	1.0 (13/10/2015)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desee desconectarse del osciloscopio/simulador</i>	
Precondición	Que el sistema esté conectado a un osciloscopio al simulador	
Secuencia normal	Paso	Acción
	1	El actor Usuario(ACT-0001) selecciona <i>Desconectar</i>
	2	El sistema <i>solicita la desconexión del osciloscopio/simulador</i>
	3	El sistema <i>informa del resultado al Usuario</i>
Postcondición	El sistema queda desconectado del osciloscopio/simulador	
Excepciones	Paso	Acción

Tabla 53- Descripción de diseño del caso de uso 008

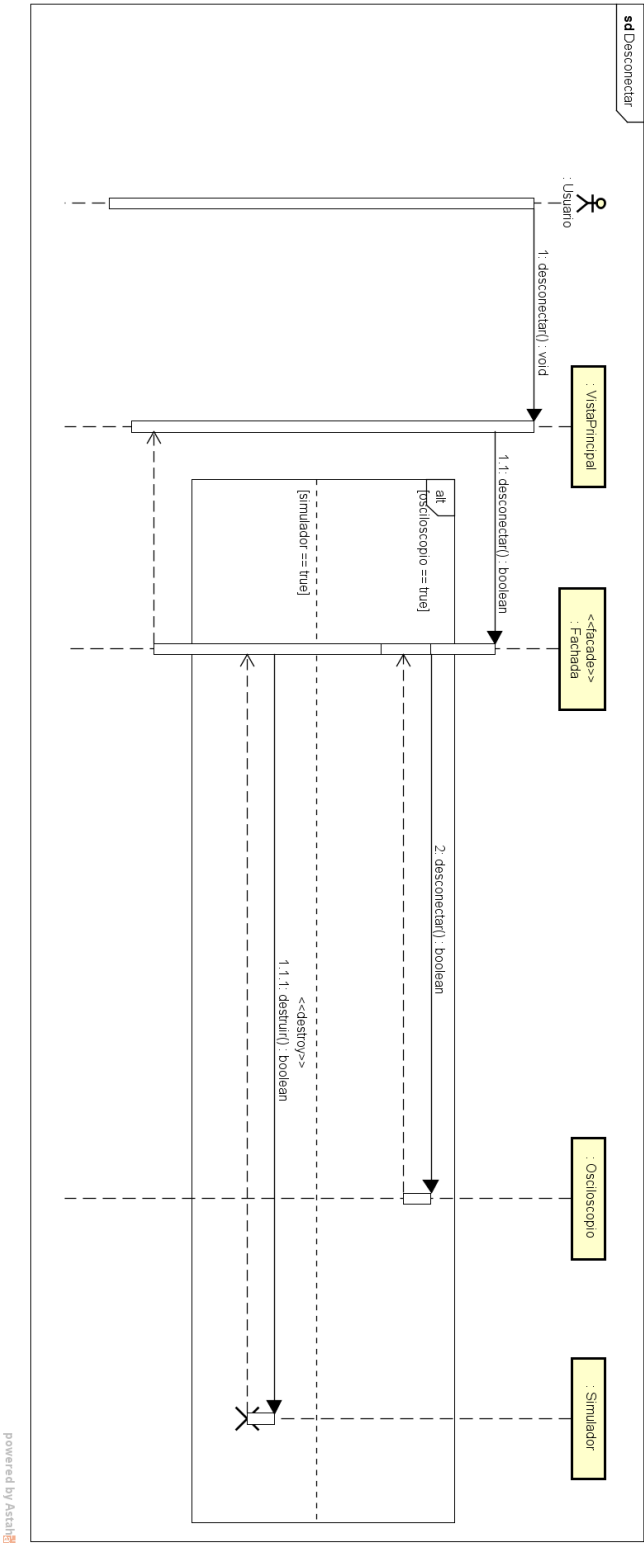


Ilustración 47-Diagrama de secuencia de diseño del caso de uso 008

UC-009: Actualizar Datos

UC-0009	ActualizarDatos	
Versión	1.0 (13/10/2015)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el Usuario desee actualizar datos</i>	
Precondición	Que el sistema esté conectado con el simulador, o osciloscopio	
Secuencia normal	Paso	Acción
	1	El actor Usuario(ACT-0001) selecciona <i>ActualizarDatos</i>
	2	El sistema <i>comprueba el canal para el cual se van a actualizar los datos.</i>
	3	El sistema <i>solicita los datos del canal al osciloscopio o al simulador.</i>
	4	El sistema <i>dibuja y muestra los datos recibidos del canal al usuario.</i>
Postcondición		
Excepciones	Paso	Acción
	3	<i>Si el sistema obtiene una respuesta negativa del osciloscopio o simulador, se informa al Usuario, a continuación el caso de uso queda sin efecto.</i>

Tabla 54- Descripción de diseño del caso de uso 009

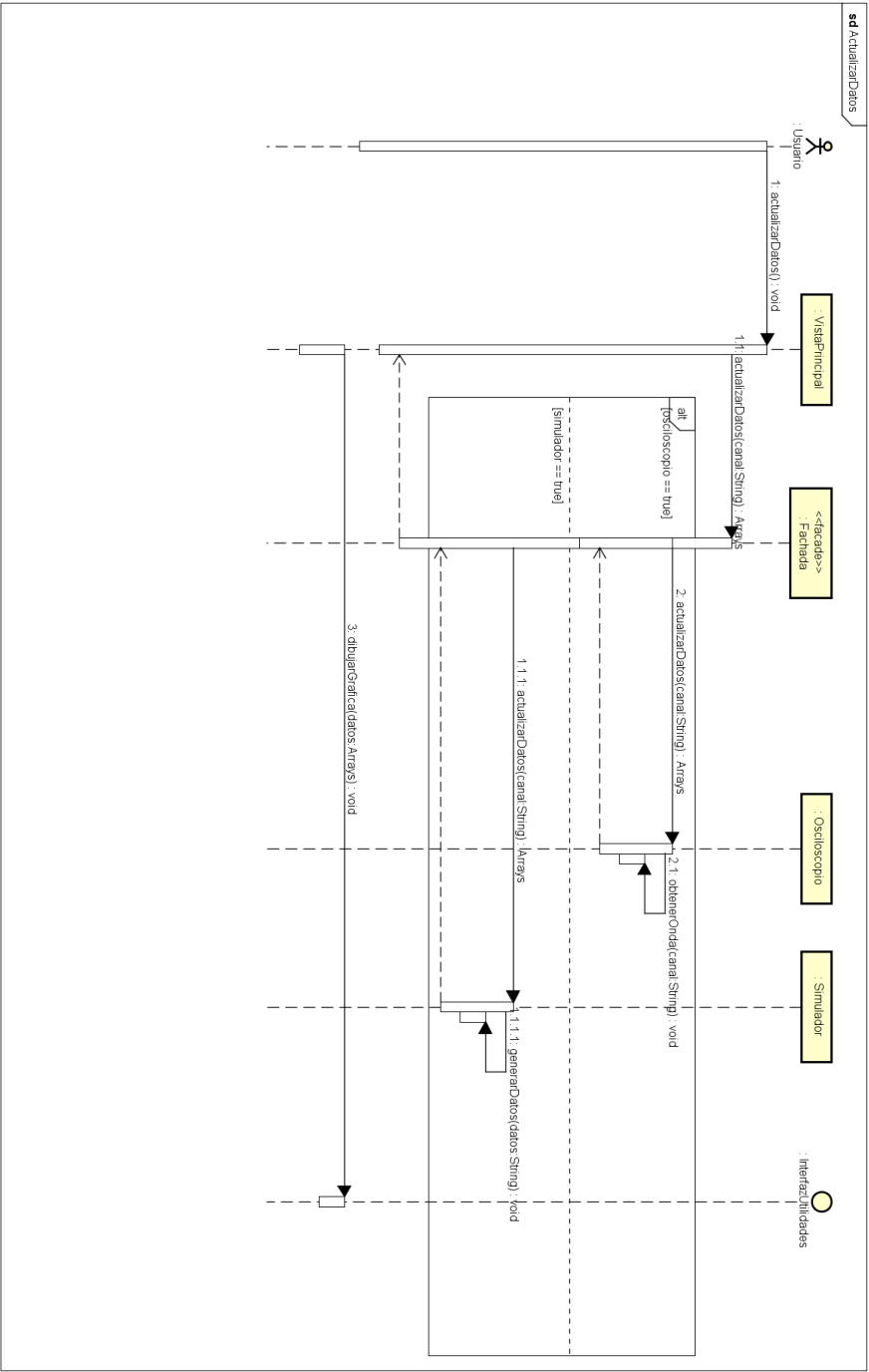


Ilustración 48- Diagrama de secuencia de diseño del caso de uso 009

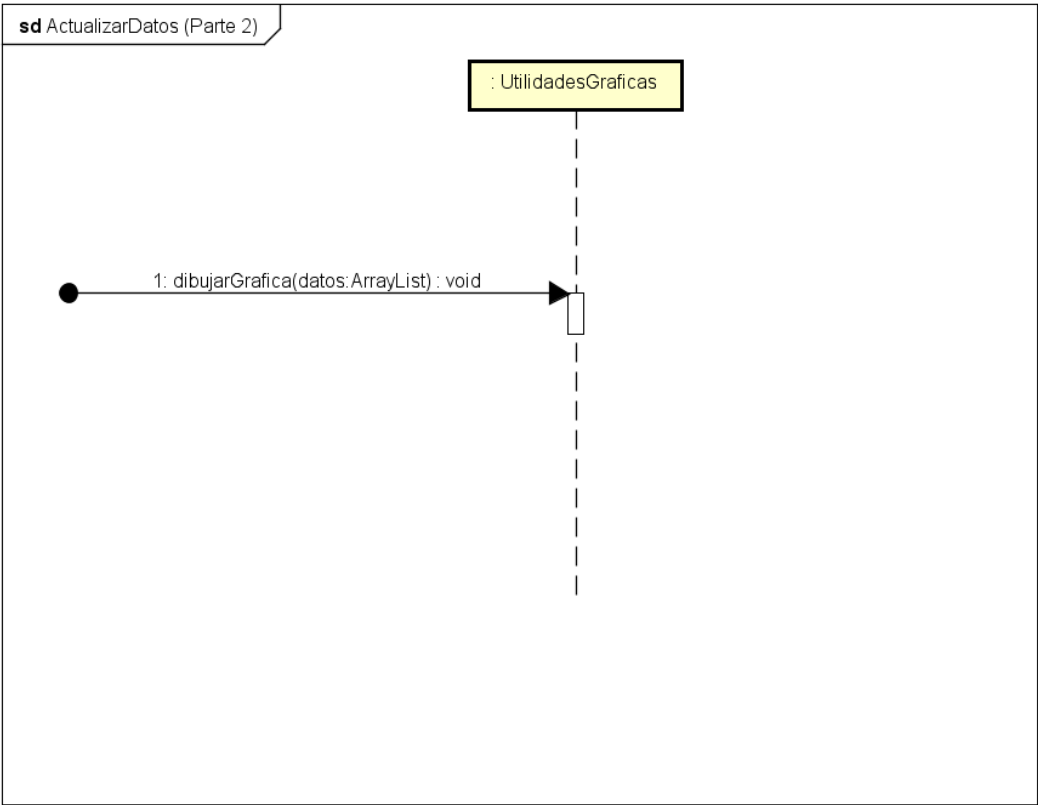


Ilustración 49- Diagrama de secuencia de diseño del caso de uso 009 (Parte dos)

UC-010: Cambiar Canal

UC-0010	CambiarCanal	
Versión	1.0 (13/10/2015)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desee cambiar el canal del osciloscopio/simulador</i>	
Precondición	Que la aplicación esté conectado al osciloscopio / simulador	
Secuencia normal	Paso	Acción
	1	El actor <u>Usuario(ACT-0001)</u> <i>selecciona Cambiar Canal</i>
	2	El sistema <i>solicita la información al osciloscopio/simulador de los canales disponibles</i>
	3	El sistema <i>muestra la lista de canales disponibles al Usuario</i>
	4	El actor <u>Usuario(ACT-0001)</u> <i>selecciona el canal a cambiar</i>
	5	El sistema <i>solicita la conexión al canal seleccionado al osciloscopio/simulador</i>
	6	El sistema <i>ha cambiado de canal.</i>
Postcondición	El sistema está conectado al nuevo canal	
Excepciones	Paso	Acción
	5	El sistema <i>no puede conectarse al nuevo canal, se invalida el cambio de canal y a continuación el caso de uso queda sin efecto.</i>

Tabla 55- Descripción de diseño del caso de uso 010

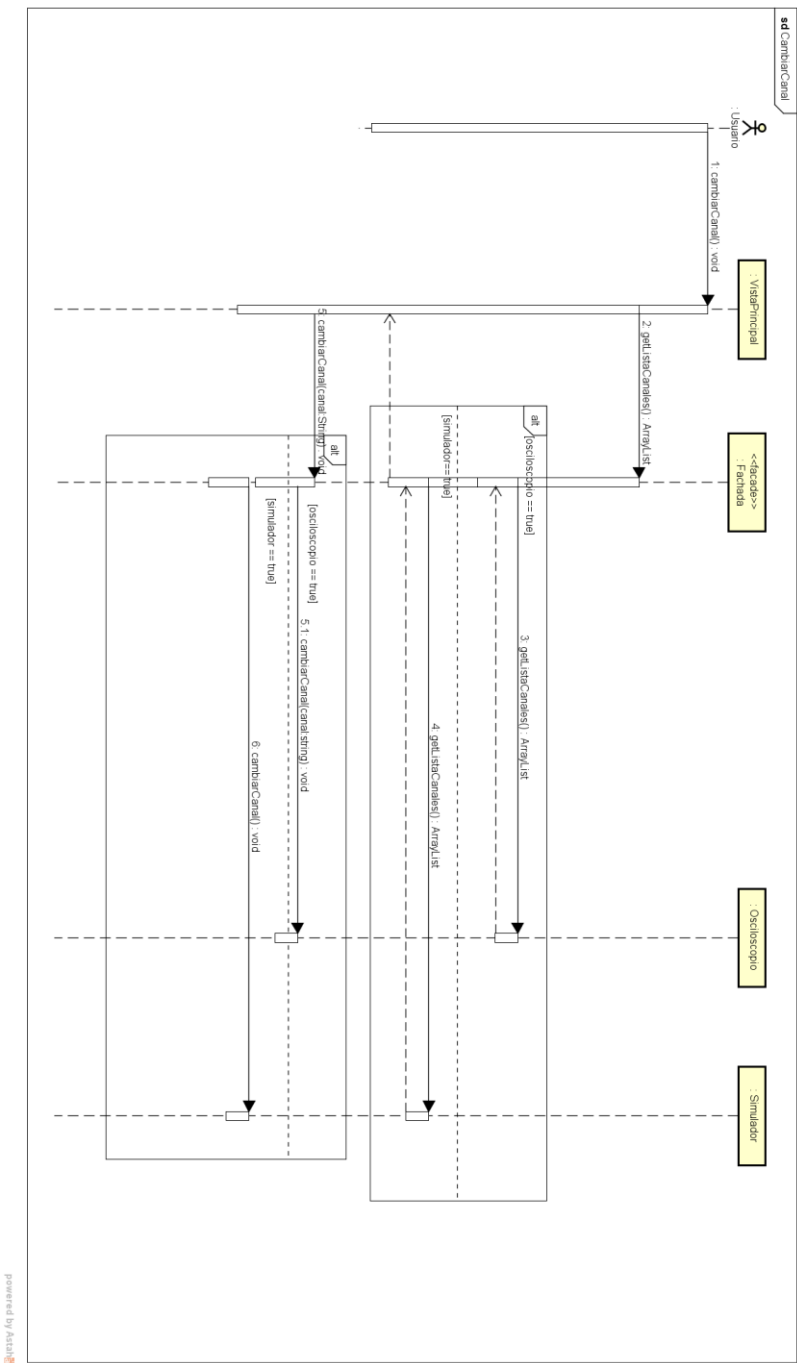


Ilustración 50- Diagrama de secuencia de diseño del caso de uso 010

UC-011: Enviar Onda

UC-0011	EnviarOnda	
Versión	1.0 (13/10/2015)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desee enviar una onda</i> .	
Precondición	Que la aplicación esté conectado al osciloscopio / simulador. Que haya una onda cargada en el sistema.	
Secuencia normal	Paso	Acción
	1	El actor <u>Usuario(ACT-0001)</u> <i>selecciona Enviar Onda</i> .
	2	El sistema <i>lee la onda cargada previamente</i> .
	3	El sistema <i>adapta el formato de la onda solicitada, al requerido por el dispositivo</i> .
	4	El sistema <i>envía la onda adaptada solicitada al dispositivo</i> .
	%	El sistema <i>informa al Usuario del resultado de la operación</i> .
Postcondición	El sistema ha enviado la onda	
Excepciones	Paso	Acción
	2	El sistema <i>no puede leer la onda cargada correctamente, se invalida el envío de onda y a continuación el caso de uso queda sin efecto</i> .
	3	El sistema <i>no puede transformar la onda al formato de datos pedido por el dispositivo, se invalida el envío de onda y a continuación el caso de uso queda sin efecto</i> .

Tabla 56- Descripción de diseño del caso de uso 011

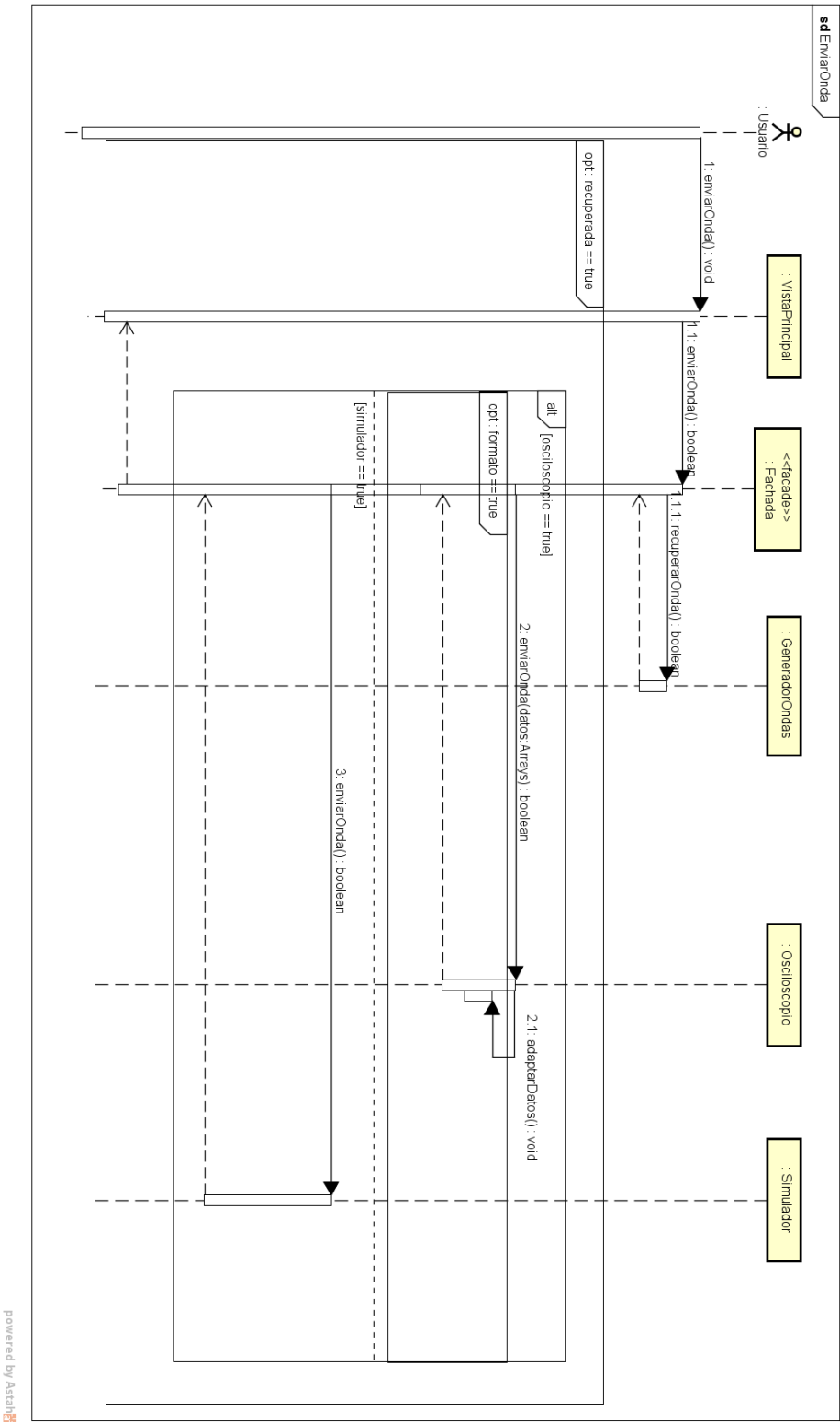


Ilustración 51- Diagrama de secuencia de diseño del caso de uso 011

5.5. Diagrama de clases de diseño

El diagrama de clases de diseño es el que se manifiesta a continuación:

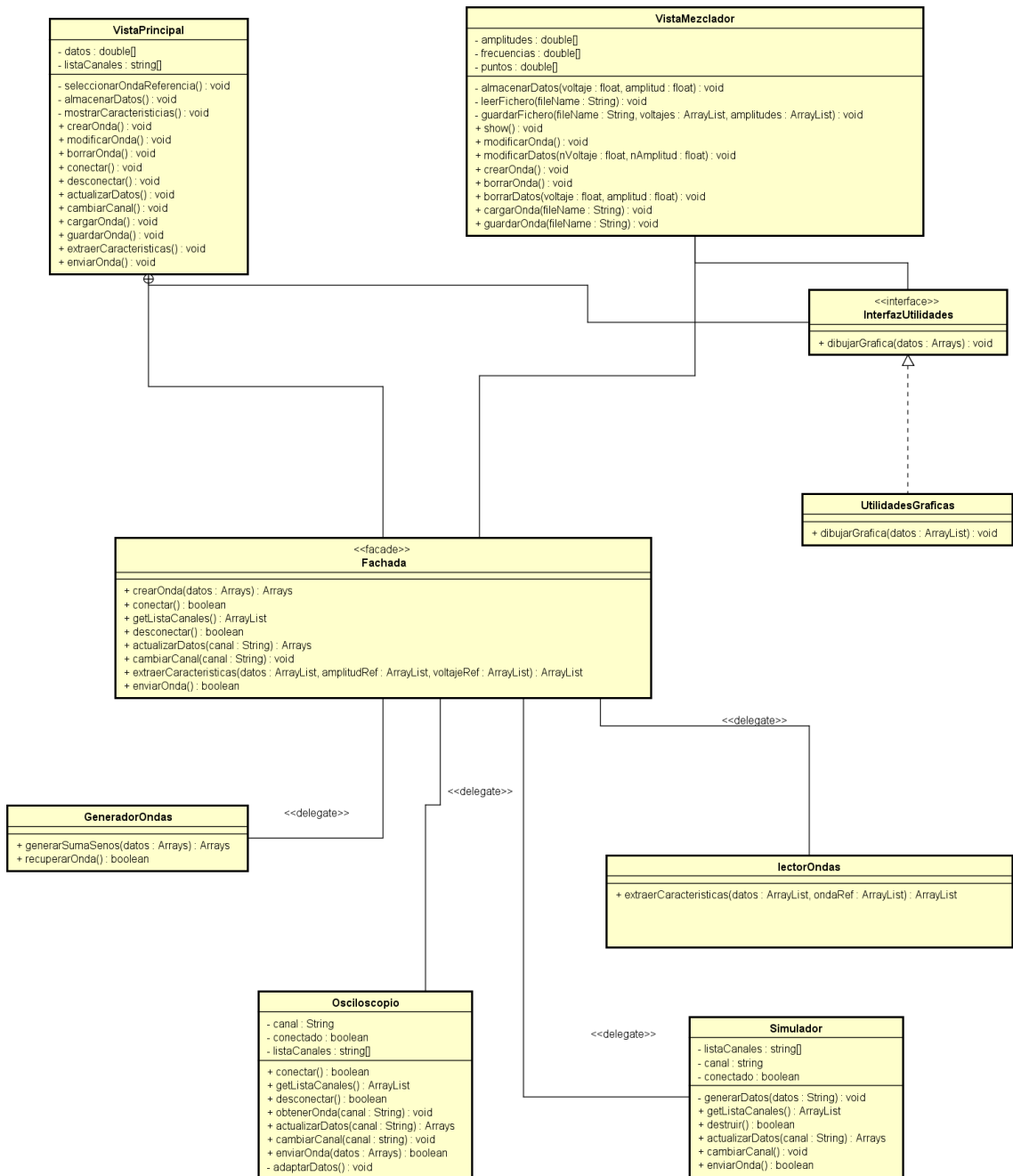


Ilustración 52- Diagrama de clases de Diseño

5.6. Diseño de la interfaz

En el siguiente apartado, se hará referencia a conocimientos técnicos correspondientes a la ingeniería de la interacción.

En esta sección se explican los principios que se han tenido en cuenta a la hora de diseñar la interfaz con la que interactúa el usuario y que conforma la aplicación, a su vez se presenta los esquemas que inicialmente se habían planteado en el proyecto de mi compañero Eduardo Rodríguez Gutiez [1], y que se han descartado.

Diseños iniciales

En este proyecto, se planteó una interfaz para la cuarta interacción similar a la ofertada por programas comerciales como “LabView” o “Proteus”. Teniendo en cuenta la experiencia deseada para los usuarios finales de la aplicación [51], resulta a priori lo más lógico imitar todo lo posible a un laboratorio real, donde se dispone de diferentes instrumentos, de distintas marcas características y con una disposición de botones distintas para cada uno de ellos:

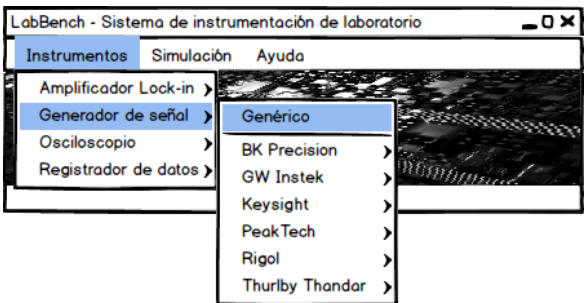
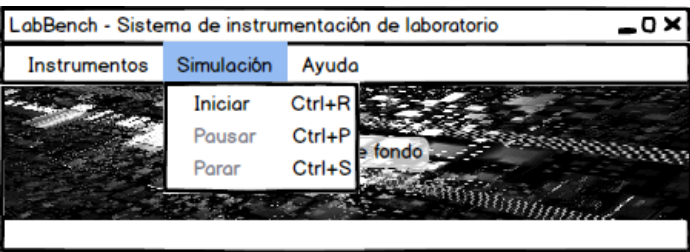


Ilustración 53 - Mock-up donde se muestra la selección de instrumentos

En este primer boceto, se seleccionan entre los posibles instrumentos, aquellos registrados por el sistema ordenados por marca mediante un menú desplegable. Inicialmente en ambos trabajos fin de master se contempló esta posibilidad, de incorporar nuevos instrumentos, pero se rechazó una implementación dado que la complejidad de la misma alejaba el objetivo principal del proyecto. Es por eso que en el capítulo 7º de esta memoria se ha planteado como mejora esto, así como indicaciones de como implementar de manera eficiente esta funcionalidad.



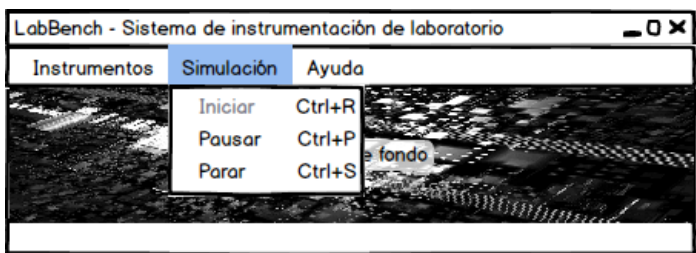


Ilustración 54 - Mock-ups donde se muestra el menú de simulación

En este segundo boceto, se presenta como sería, el inicio de conexión con los distintos dispositivos y del experimento, así como de la captura y interpretación de los datos, se planteó como un menú desplegable, etiquetado bajo el menú de simulación, donde mediante tres acciones el experimento se podría “Iniciar”, “Pausar” ó “Parar”.

Las razones para descartar esta idea fueron:

- La etiqueta “**Simulación**” puede llevar a error al confundirlo con el uso del simulador en vez del osciloscopio.
- En caso de requerirse una acción rápida por parte del usuario para parar el experimento, esta opción según la ley de Fitts que mide el tiempo en alcanzar un objetivo determinado en la pantalla, es bastante mayor a otras opciones como puede ser un menú contextual o un simple botón permanente en la interfaz.
- Existe un principio de diseño de interfaces que menciona que no es aconsejable tener opciones peligrosas juntas. En este caso, si el usuario desea “**Pausar**” un experimento, es alta la probabilidad de que pulse accidentalmente “**Parar**”, forzando a reiniciar el experimento por completo.

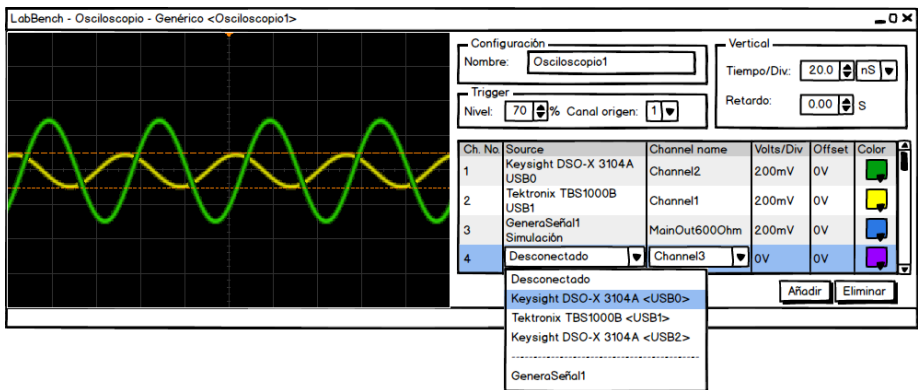


Ilustración 55 – Mock-up de la vista del osciloscopio genérico

En este tercer mock-up, se definió una vista genérica para opción de usar múltiples osciloscopios, recibiendo uno por cada canal, siendo posible añadir canales o eliminarlos según fuera necesario. A su vez los distintos canales y los datos recibidos se podrían etiquetar por color. Las razones para descartar este boceto fueron las siguientes:

- El sistema a priori solo puede estar ejecutándose a la vez sobre un único tipo de instrumento, por lo que no tiene sentido añadir nuevos canales.
- El número de canales disponibles viene dado por el instrumento en sí, por lo que hacer una implementación de estas características requería añadir una asociación entre el canal “**virtual**” de la aplicación y el canal al que se desea conectar del instrumento.

Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama

- La opción de “**Añadir**” está físicamente conectada a la de “**Eliminar**” lo que podría provocar errores por parte del usuario.

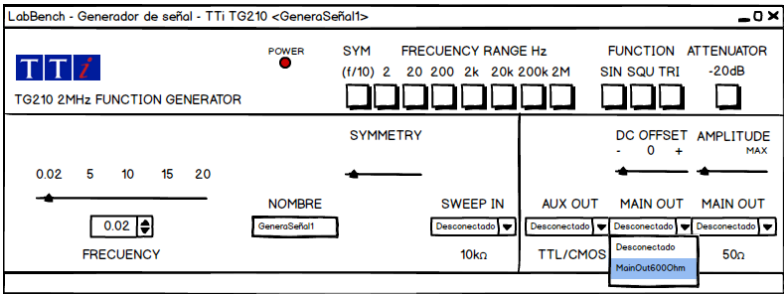


Ilustración 56-Mock-up donde se muestra la interfaz del generador de señal genérico

Esquema inicial para el generador de frecuencias. Se utiliza una metáfora para simular el aspecto del generador como si fuera el real. Existen casos donde se demuestra que el uso de metáforas en el diseño de este tipo de aplicaciones es erróneo y provoca una mala experiencia de usuario. A mayores citar que el tipo de onda generada se limita por requisitos a onda sinusoidal, por lo que carece de sentido mantener todas las opciones disponibles del instrumento. En la interfaz final, se mantuvo eso sí los controles tipo deslizador para mejorar la experiencia del usuario.

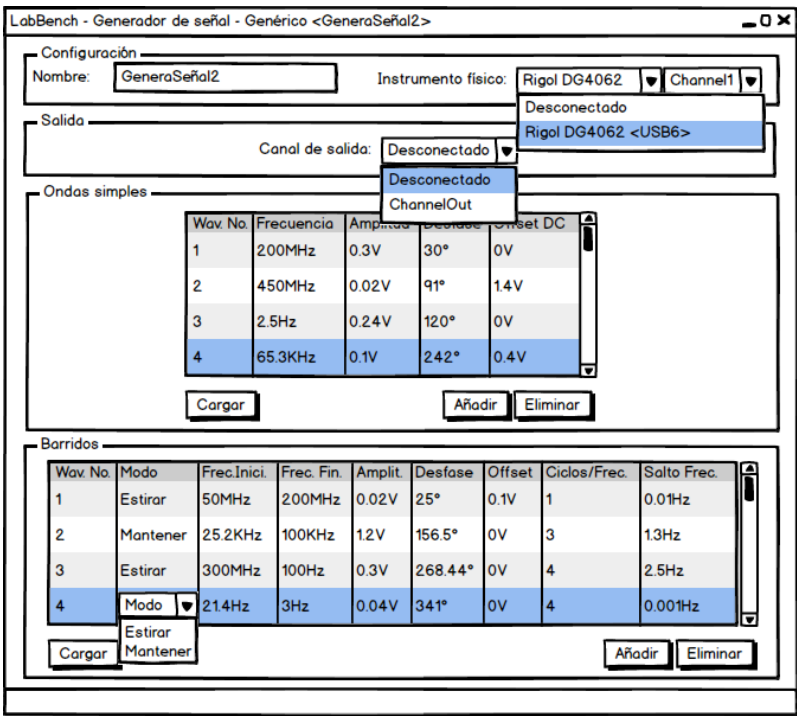


Ilustración 57- Mock-up del generador de ondas

El quinto mock-up, hace referencia al oscilador genérico. Gran parte de esta interfaz se utiliza en la versión final. Donde se han eliminado diferentes opciones (offset, barrido...) y se ha añadido una gráfica a modo de vista previa para poder visualizar los datos generados.

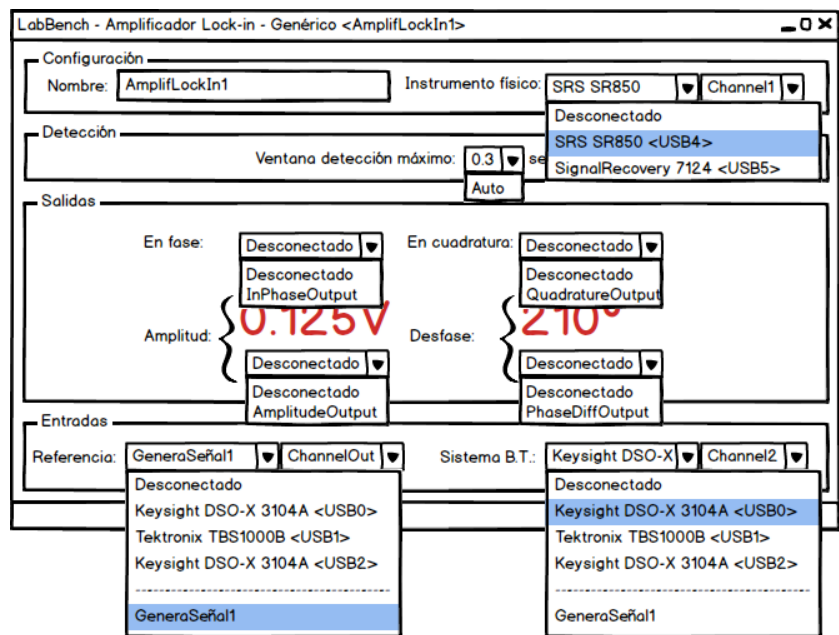


Ilustración 58 - Mock-up del amplificador lock-in

El sexto mock-up proporciona un sistema de acceso sencillo al lock-in genérico. Dado que la conversión de datos de digital a analógico escapa al alcance de este proyecto, se plantea como futura mejora.

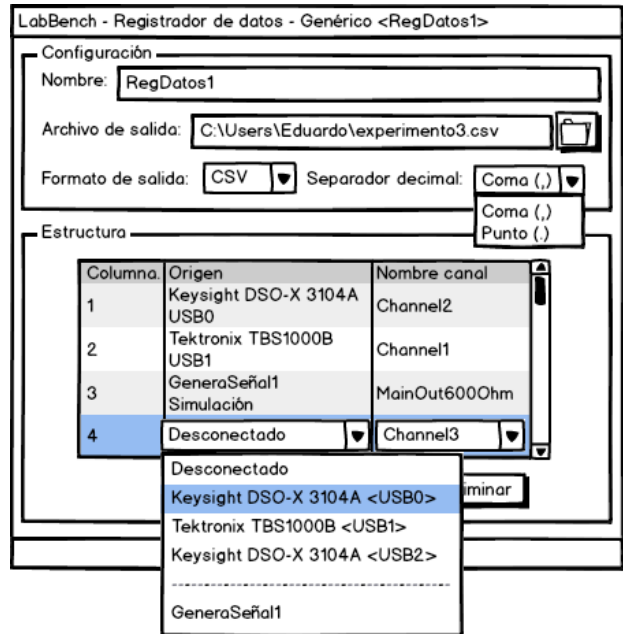


Ilustración 59 - Mock-up del registrador de datos

Este sexto mock-up, muestra una avanzada forma de guardar los datos. La idea inicial era poder crear un instrumento virtual que actuase como registrador de datos del experimento, para ello el mismo permitiría no solo elegir el formato de salida o el directorio, sino que canales y con que nombre se iban a guardar.

Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama

Por simpleza de la aplicación y no alejarse del objetivo principal se optó por guardar los datos de cada canal por separado y en formato de texto plano.

En el caso del mezclador de ondas, se podría guardar los pares de voltaje y amplitud que conforma las ondas en formato plano o en XML.

Diseño Final

A continuación, se muestra la interfaz final de la aplicación, así como los motivos desde el punto de vista de la ingeniería de la interacción que han llevado a implementar este modelo.

Las líneas generales en función de la usabilidad, para el desarrollo de la interfaz han sido [51]:

- **Imitar el aspecto de la interfaz “MetroUI” [52] de Microsoft, lo que implica:**
 - Utilizar como fuente principal *Segoe UI*.
 - Los botones y “*widgets*” han de tener un único color de fondo y el texto de color blanco y centrado.
 - Colores simples de fondo para los botones, en concreto se han utilizado el rojo, verde y azul en base al estándar HTML.
- **Facilidad de aprendizaje:**
 - Se han utilizado, frases completas y verbos de acción en lugar de iconos para expresar las tareas que se pueden realizar.
 - Se han utilizado “*tooltips*”, como ayuda complementaria a la aplicación.
- **Facilidad de recuerdo:**
 - Se han utilizado distintos colores para tareas que no guardan a priori ninguna relación. (Ej: “Enviar onda” y “Guardar Datos”.
 - Se ha unido toda la funcionalidad de la aplicación, excepto el generador de ondas y extraer características en una misma ventana, facilitando el recuerdo del usuario para buscar la información que necesita.
- **Tolerancia al error:**
 - Se ha separado acciones peligrosas entre si dentro de la aplicación.
 - Acciones peligrosas, tienen colores diferentes entre si.
 - Se deshabilitan opciones si estas entran en conflicto con alguna otra (Ej: Elegir conectar entre el simulador o el osciloscopio cuando ya se ha realizado una conexión), marcando además las opciones deshabilitadas con un color gris.

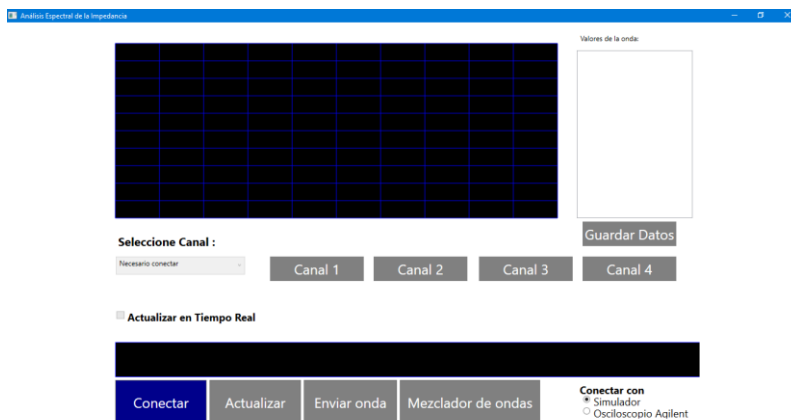


Ilustración 60- Interfaz de la aplicación, nada más ser ejecutada

Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama

Captura de la vista principal. En ella se muestran como únicas opciones disponibles, “*Conectar*” y con qué tipo de dispositivo, el Simulador o el Osciloscopio Agilent.

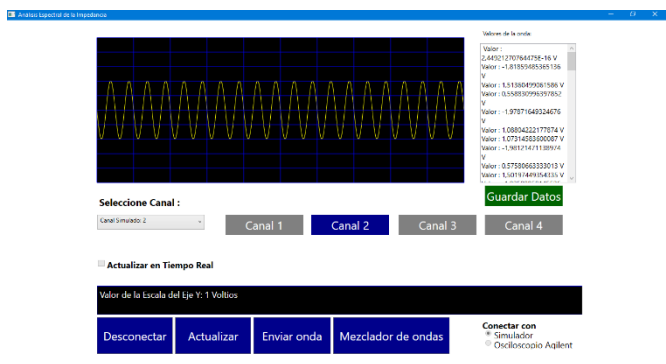


Ilustración 61- Ventana principal de la aplicación con una onda recibida desde el canal 2

Una vez conectado, se permite elegir entre los distintos canales posibles, mediante el desplegable “*Seleccione Canal*” y a su vez entre las siguientes opciones:

- **Desconectar:** Desconecta del instrumento seleccionado previamente.
- **Actualizar en tiempo real:** Marcando este checkbox, el osciloscopio o el simulador envían automáticamente de manera periódica nuevos datos al sistema.
- **Actualizar:** Realiza una captura de los datos del instrumento seleccionado de manera asíncrona.
- **Enviar Onda:** Envía la onda creada en el “*Mezclador de Ondas*” al instrumento para que la genere. Esta opción está deshabilitada si no se ha creado ninguna onda con el mezclador.
- **Guardar Datos:** Guarda los puntos de voltaje en punto flotante, en un archivo de texto plano a elección del usuario.

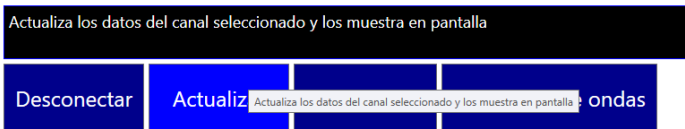


Ilustración 62- Descripción de ayuda para el botón “Actualizar”

Esta vista es de la misma aplicación, y simplemente muestra entre los distintos tipos de ayuda con los que el usuario puede contar.

Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama

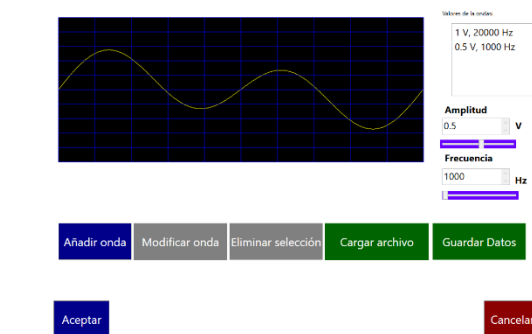


Ilustración 63- Interfaz final del mezclador de ondas

Vista del mezclador o generador de ondas. Las ondas creadas son mostradas en el “**listBox**” titulados “*Valores de las ondas*”. Las opciones disponibles son:

- **Añadir Onda:** Añade una nueva onda al sistema, a partir de los valores de amplitud y frecuencia.
- **Modificar Onda:** Actualiza, los valores de una onda seleccionada desde “*Valores de las ondas*” con los valores introducidos en amplitud y frecuencia.
- **Eliminar Selección:** Elimina una onda desde “*Valores de las ondas*” y a continuación actualiza los datos.
- **Cargar archivo:** Carga un archivo con pares de amplitud y frecuencia y a continuación los inserta en “*Valores de las ondas*”.
- **Guardar Datos:** Guarda un archivo con los pares de amplitud y frecuencia que están en “*Valores de las ondas*”.
- **Aceptar y Cancelar:** Ambas vuelven a la vista principal de la aplicación, para mejorar la tolerancia a fallas ambas tienen un color diferente y están separadas.
- **Amplitud y frecuencia:** Introducen los valores de amplitud en Voltaje y de frecuencia en Hertzios para generar la onda sinusoidal deseada, para mejorar la experiencia del usuario se han implementado dos modos de introducir los datos:
 - Introduciendo los valores en el “**textbox**” directamente.
 - Utilizando los “**sliders**” suministrados.

6. Capítulo sexto: IMPLEMENTACIÓN DE LA APLICACIÓN

En este capítulo, se explican los detalles del desarrollo de las distintas iteraciones, así como se han resuelto algunos de los problemas encontrados en las mismas.

- Implementación de la interfaz
- Generador de onda arbitraria
- Medida de corrientes de bajo nivel
- Comunicación con el osciloscopio

6.1. Primera Iteración

En un momento inicial, se aconsejó por parte del cliente, utilizar LabView [5] como herramienta de desarrollo para la aplicación.

LabView fue descartado por las siguientes razones:

- Desconocimiento de la plataforma, por parte del equipo de desarrollo.
- Paradigma gráfico bajo el “Lenguaje G”, qué limita las posibilidades de desarrollo de funcionalidades avanzadas tales como la adaptabilidad a nuevos sistemas.
- Falta de funcionalidad (en concreto para controlar el generador de ondas), en los drivers proporcionados por Agilent.

La siguiente opción para el desarrollo fue el lenguaje Matlab [53], este era conocido por ambos desarrolladores debido a que su uso se imparte en la asignatura del Máster llamada ‘Métodos avanzados de razonamiento y representación del conocimiento’.

Como suite inicial, de las proporcionadas por Matlab se optó por la proporcionada por el cliente, en este caso la versión R2013b, instalando no solo la suite sino también las herramientas de ‘Signal Processing Toolbox’ y ‘Instrument Control Toolbox’. Posteriormente se descargaron e instalaron los drives del osciloscopio/generador bajo el nombre de ‘2000, 3000, 4000, 6000 InfiniiVision X-Series Oscilloscope IVI and MATLAB Instrument Drivers’ disponibles desde la página de Keysight.

Adicionalmente, también es necesario instalar las siguientes librerías:

- **‘IVI Shared Components’**: Contiene la funcionalidad común a todos los controladores que implementan los estándares IVI. [54]
 - **Página Web**: http://www.ivifoundation.org/shared_components/
- **‘KeySight Connection Expert’**: Permite obtener los identificadores de los dispositivos de la marca Keysight/Agilent conectados al ordenador. Se instala como parte del paquete IO Library Suite.

- **‘IO Library Suite’:** Ofrece diversos servicios a los dispositivos Keysight/Agilent conectados al ordenador. Entre ellos monitorización, envío y lectura de comandos GPIB recibidos por el equipo etc... [55]
 - **Página Web:** <http://www.keysight.com/en/pd-1985909/io-libraries-suite?cc=ES&lc=eng>
- **Adicional: ‘Benchlink Wave Builder Pro’:** Permite crear y enviar formas de onda arbitrarias al generador de ondas. Este programa ha sido de gran utilidad en el desarrollo para comprender el funcionamiento del osciloscopio, de cara a la generación de ondas con el fin de mediante ingeniería inversa adaptar la aplicación. [56]
 - **Página Web:** <http://www.keysight.com/en/pd-1962285-pn-33503A/benchlink-waveform-builder-pro-software?cc=ES&lc=eng>

La primera versión que se realizó del software, tal y como se ha citado anteriormente, se realizó en Matlab. En esta iteración se intentó explorar las librerías, y su uso, descubriendo por ejemplo que, aunque se funcione por USB, en realidad se está enviando comandos GPIB debido a que las librerías son en realidad una serie de ‘Wrappers’ que transforman las llamadas al citado lenguaje que se envían mediante transferencia USB al osciloscopio.

Para el desarrollo de esta primera iteración se utilizó el ejemplo de Matlab proporcionado por el fabricante y que se puede encontrar en la siguiente ruta del disco duro de instalación: “*C:\Program Files\IVI Foundation\IVI\Drivers\AgInfiniiVision\Examples\MATLAB*”. Modificando el código anterior, se logró configurar el osciloscopio para capturar de forma continua los valores medidos a través de los cuatro canales del DSO-X 3104A.

Posteriormente, a través de diversos ejemplos de la comunidad de Matlab se logró dibujar los datos obtenidos de los canales del dispositivo. Finalmente usando la ayuda a la programación de Matlab se logró descubrir que las funcionalidades del generador de ondas situadas como propiedades en el objeto “*driver*” no estaban implementadas, aunque la firma de la operación estuviera presente para su llamada. Esto finalmente llevó a descartar el código en Matlab y buscar entre los distintos “*wrappers*” del driver IVI cual tenía implementada la funcionalidad del generador de ondas.

Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama

El código de esta iteración, basado también en parte en [57] ,se puede ver en el siguiente listado:

```
%Obtenerdatosdelosciloscopio
%osc=oscilloscope();
%targets=getResources(osc);
%targets
%set(osc,'Resource','USB0::0x0957::0x17A0::my53XXXXX::0::INSTR');
%connect(osc);
%%mex-setup
%%connect(osc);
%data=getWaveform(osc);
%plot(data);
clc
%A partir de aqui cargamos el driver bueno
try
    %Createdriverinstance
    driver=instrument.driver.AgInfiniiVision();
    %Edit resource and options as needed. Resource is ignored if option Simulate=true
    resourceDesc='USB0::0x0957::0x17A0::my53XXXXX::0::INSTR';
    initOptions='QueryInstrStatus=true,Simulate=true,DriverSetup=Model=,Trace=false';
    idquery=true;
    reset=false;

    driver.Initialize(resourceDesc,idquery,reset,initOptions);
    disp('DriverInitialized');
    disp(blanks(1));
    %Comando para parar la medicion
    %driver.DeviceSpecific.Measurements.Initiate();
    %Llamamos al generador de funciones
    generador=driver.DeviceSpecific.WaveformGenerator;
    generador.Function='AgInfiniiVisionWaveformGeneratorFunctionSinusoid'
    generador.OffsetVoltage=-2.5;
    generador.Frequency=2;
    generador.Amplitude=5.0;
    whitebg('black');
    clear title xlabel ylabel
    hf=figure('position',[00epseps],'menubar','none');
    while(true)

        [WaveformArray,InitialX,XIncrement]=driver.DeviceSpecific.Measurements.Item('Channel1'
    ).ReadWaveform(inf);

        [WaveformArray2,InitialX,XIncrement]=driver.DeviceSpecific.Measurements.Item('Channel12
    ').ReadWaveform(inf);
        subplot(2,2,1);
        plot(WaveformArray,'y');
        clear title xlabel ylabel
        title='Canal1';
        subplot(2,2,2);
        plot(WaveformArray2,'g');
        clear title xlabel ylabel
        title='Canal2';
        if strcmp(get(hf,'currentcharacter'),'q')
            close(hf)
            break
        end
    drawnow
    end
    catch exception
        disp(getReport(exception));
    end
    if driver.Initialized
        driver.Close();
        disp('Driver Closed');
    end

    disp('Done')
    disp(blanks(1));
```

Tabla 57- Código de la primera iteración

Los primeros comentarios comentados permiten conocer la lista de dispositivos conectados al ordenador.

En este código comentado, se crea un objeto del tipo “*Oscilloscope*” y se le asigna a osc.

Una vez creado el objeto, se llama a la función “*getResources*” que devuelve la lista de dispositivos conectados almacenándolos en la variable “*Targets*”. A continuación, se muestra la variable citada anteriormente, para ver los identificadores de los dispositivos conectados.

Los identificadores de los dispositivos, son cadenas de caracteres con el formato que aparece en las líneas 5 y 19 del código presentado, son de la forma USB0::0x0957::0x17A0::my53XXXXX::0::INSTR, con campos separados por dobles dos puntos. El número de campos y su significado varía en función del tipo de conexión. Los campos representan los valores indicados a continuación:

- Primer campo: Indica el modo de conexión, puede ser USB, GPIB o TCPIP en caso de estar conectado por internet.
- Segundo campo: VID o vendedor del dispositivo, en caso de estar conectado por internet este campo se convierte en la dirección IP, y se elimina en el caso de GPIB.
- Tercer campo: El PID USB asignado al dispositivo, en caso de estar conectado vía red este campo se elimina.
- Cuarto campo: El número de serie del dispositivo.
- Sufijo: 0 e INSTR, ambos inmutables pese al cambio de conexión.

Además de éste código, se desarrolló algunas funciones adicionales que permitieran sintetizar ondas con solo una llamada, y funciones de dibujo.

6.2.Segunda Iteración

En el comienzo de la segunda iteración, como se ha citado anteriormente se decidió cambiar el desarrollo a C# por la imposibilidad de implementar los requisitos en Matlab debido a la falta de soporte por parte del fabricante.

C# como ya se ha comentado en la memoria fue utilizado en anteriores desarrollos por el equipo por lo que el aprendizaje del mismo no era necesario. Existe, al igual que con Matlab, un código de ejemplo en C# proporcionado por el fabricante para el desarrollo, dicho ejemplo se encuentra en la siguiente ruta:

“*C:\Program Files\IVI Foundation\IVI\Drivers\AgInfiniiVision\Examples\CSharp\CsExample1*”

Se crea diversos proyectos en Visual Studio para ambos miembros del equipo, suficientes para crear un esquema de clases básico y escribir código en el.

Uno de estos proyectos, se denominó ‘*SimuladorOndas*’, cuya función era proporcionar un generador de ondas y simular su resultado mostrándola. Los casos de uso implementados por esta iteración fueron los siguientes:

- Crear Onda
- Modificar Onda
- Borrar Onda
- Cargar Onda
- Guardar Onda

La interfaz de usuario, que luego se heredaría al proyecto final es la que se puede apreciar en la siguiente imagen:

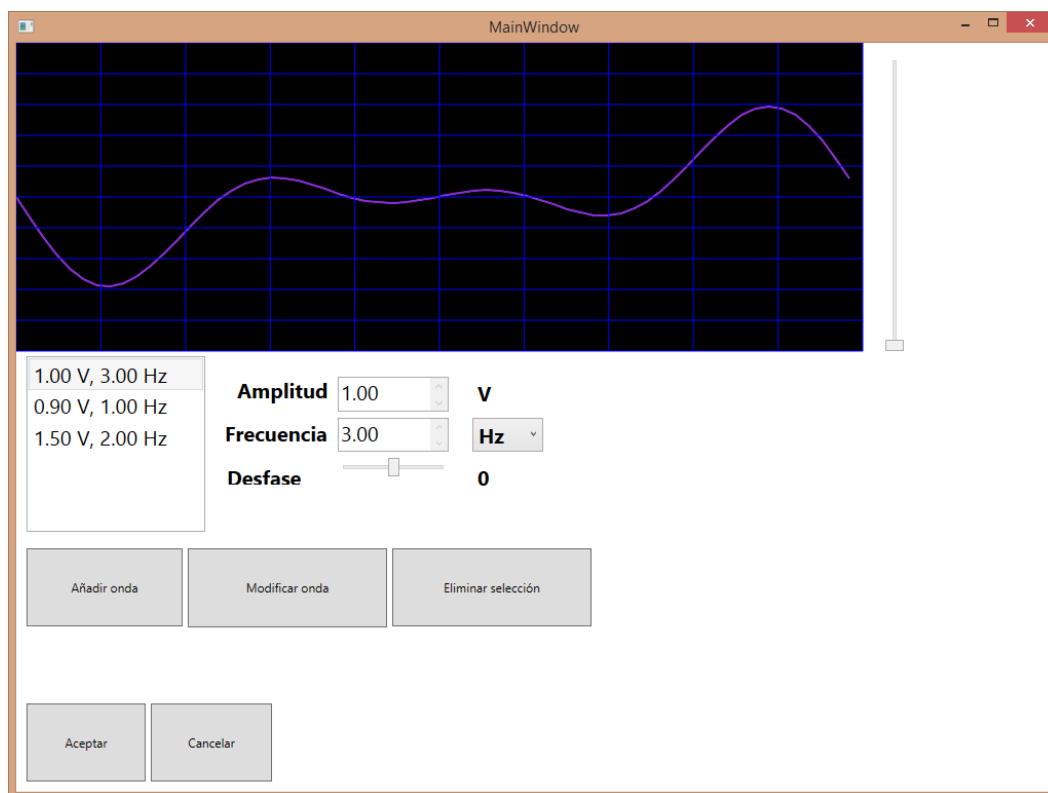


Ilustración 64 - Aspecto de la interfaz de la segunda iteración

Como se puede apreciar, los sliders y textboxes de introducción de datos estaban implementados, pero aún no estaban en su disposición final, ni con distintos colores para mejorar la experiencia del usuario. A mayores había inicialmente un slider vertical situado a la derecha de la gráfica que permitía aumentar o disminuir la escala. Este slider se eliminó en siguientes iteraciones debido a que interfería con el osciloscopio y proporcionaba distorsión a la onda dibujada.

El diagrama de clases, para esta iteración, extraído de [1], es el de la siguiente imagen:

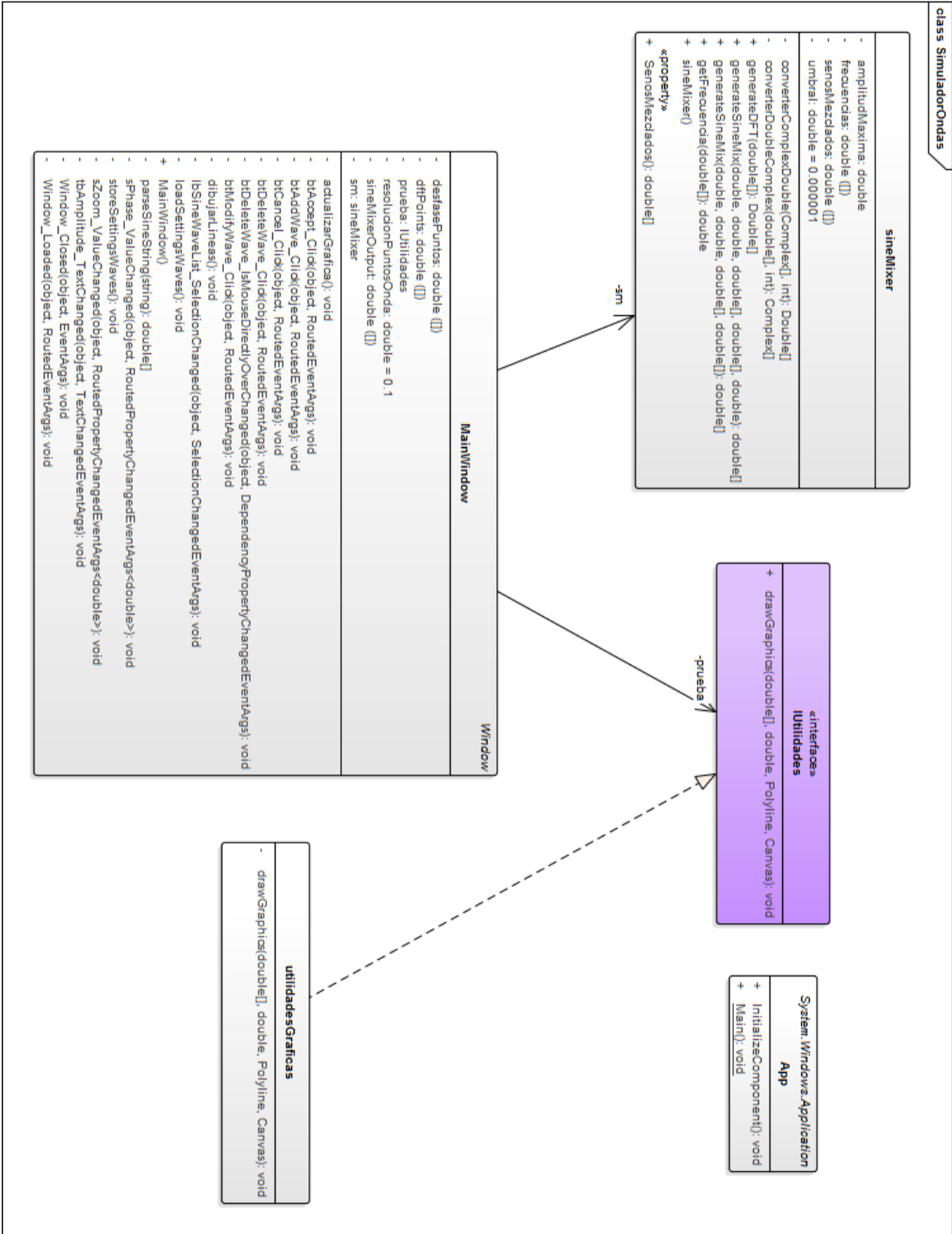


Ilustración 65- Diagrama de clases de la segunda iteración

Este diagrama tiene tres clases:

- **MainWindow:** Controla la vista principal, y los métodos de dibujo de la señal.
- **SineMixer:** Genera a través de la función sobrecargada: “*generateSineMix*”, el vector de voltajes resultado de sumar las ondas definidas. En esta iteración la forma de calcular las ondas no era la correcta, puesto que aún no se habían tenido en cuenta la escala de los datos impuesta por el osciloscopio, por lo que los resultados oscilaban en un intervalo de amplitudes de [-5V, 5V]. Finalmente se apreció que era preferible encapsular las ondas como un objeto, así como sus valores de amplitudes y voltajes, e implementarlo en la siguiente iteración.
- **UtilidadesGraficas:** Al igual que lo explicado en el capítulo anterior en esta versión las utilidades graficas ayudan al dibujado de la señal en la interfaz, eliminando esa tarea del hilo principal de ejecución.

Finalmente citar, que en esta iteración ya se almacenaban y recuperaba las preferencias del usuario. implementando esta funcionalidad con el objeto “*MySettings*”, objeto por defecto de .NET que se encarga de esta tarea. De esta manera, la lista de ondas se salva al cerrar la aplicación y se vuelve a recuperar de nuevo al iniciar la aplicación.

El final de esta iteración vino marcado cuando se comprobó que el código de los distintos proyectos estaba sin refactorizar y con poca lógica en la distribución de las clases, esto llevó a una refactorización general, converger todo en un único proyecto y por tanto una nueva iteración.

6.3. Tercera Iteración

La tercera iteración del proyecto, comenzó como se ha citado antes con la refactorización y unión de los proyectos anteriores. Para ello se creó en Visual Studio, un nuevo proyecto llamado “*ProyectoRefactorizado*”, en el que se rediseño el esquema de clases, al presentado en el capítulo quinto.

6.3.1. Proyecto ‘ProyectoRefactorizado’

En este proyecto, como se ha citado anteriormente, se realizó los ajustes necesarios en implementación para que estuviera acorde a lo presentado en el capítulo quinto de esta memoria. En esta iteración se lograron implementar los siguientes casos de uso:

- *Enviar Onda.*
- *Actualizar Datos.*
- *Conectar con el osciloscopio/simulador.*
- *Cambiar Canal.*
- *Desconectar con el osciloscopio/simulador.*

A continuación, se explicará para aquellos casos de uso implementados que se consideren de interés, el cómo se ha realizado y las dificultades encontradas.

6.3.1.1. Enviar Onda

En esta iteración, se logró conectar el osciloscopio con la aplicación y mandarle la onda de suma al DAC y recibir de vuelta los valores de voltaje.

Este caso de uso fue el que más defectos ha causado a la hora de la implementación debido a la falta de documentación del fabricante sobre el formato de datos esperados y la hora de interpretarlo el generador.

Como se ha citado antes, el generador de ondas del osciloscopio, recibe ondas arbitrarias en el modo *'Arbitrary Waveform'* en formato de voltaje en punto flotante que se envían desde la aplicación usando para ello dos posibles funciones, *'setData()'* y *'setDataDAC()'*. La primera *'setData()'* recibe los datos en punto flotante entre 0 y 1 y los datos están mapeados al rango de voltajes del generador de ondas. Es decir, $[0,1] \rightarrow [-2.5V, 2.5V]$. En cambio *'setDataDAC()'* trabaja con números enteros comprendidos en el intervalo entre $[0, 255]$.

Finalmente y dado que el rango de voltaje que se iba a usar iba estar discretizado en números con punto flotante se utilizó la función *'setData()'*, no obstante quedaban dos preguntas pendientes, respecto al cambio de formato, la primera era como se iba a obtener la frecuencia y la segunda como convertir de nuevo las amplitudes al valor correcto.

Para entender este apartado es preciso, en primer lugar, revisar el código realizado para sumar dos o más ondas.

Cálculo de la suma de ondas

Para el cálculo de la suma de ondas, se creó la siguiente función:

```
public double[] generarSumaSenos(double resolucion, double ciclos, double[] amplitudes, double[] frecuencias)
```

Esta función recibe los siguientes parámetros:

- **Resolución y ciclos:** Permiten delimitar el número de puntos que se van a enviar al osciloscopio.
- **Amplitudes:** Vector con las amplitudes de las ondas creadas
- **Frecuencia:** Vector con las frecuencias de las ondas creadas
- **sumaSenosArray:** Vector resultado con la suma de las ondas.

El código que gestiona la función es el siguiente:

```
public double[] generarSumaSenos(double resolucion, double ciclos, double[] amplitudes, double[]
frecuencias)
{
    int numeroPuntosTotales = (int)Math.Truncate(((ciclos * Math.PI) / resolucion) + 1);
    double[] calculoResolucionArray;
    double[] sumaSenosArray;
    double[] frecuenciasCopia = new double[frecuencias.Length];
    frecuencias.CopyTo(frecuenciasCopia, 0);
    calculoResolucionArray = new Double[numeroPuntosTotales];
    sumaSenosArray = new Double[numeroPuntosTotales];

    //Crear el vector y = x, con una resolucion dada por parametro
    for (int i = calculoResolucionArray.GetLowerBound(0); i
calculoResolucionArray.GetUpperBound(0) + 1; i++)
    {
        calculoResolucionArray[i] = (resolucion * i);
    }
    for (int i = amplitudes.GetLowerBound(0); i < amplitudes.GetUpperBound(0) + 1; i++)
    {
        for (int j = calculoResolucionArray.GetLowerBound(0); j <
calculoResolucionArray.GetUpperBound(0) + 1; j++)
        {
            if (frecuenciasCopia[i] >= 10)
            {
                while (frecuenciasCopia[i] >= 10)
                {
                    frecuenciasCopia[i] /= 10;
                }
            }

            sumaSenosArray[j] += amplitudes[i] *Math.Sin((frecuenciasCopia[i] *
calculoResolucionArray[j]));
        }
    }
    return sumaSenosArray;
}
```

Tabla 58- Código correspondiente a sumar dos ondas

La descripción del algoritmo es la siguiente: Inicialmente se calculan los puntos que se van a crear y por lo tanto el tamaño de “sumaSenosArray”. Después se crea un vector con las coordenadas de estos puntos, este es “calculoResolucionArray” que tiene el mismo tamaño que el vector resultado. Una vez realizado esto, en cada posición del array resultado se ejecuta la siguiente sentencia:

```
sumaSenosArray[j] += amplitudes[i] *Math.Sin((frecuenciasCopia[i] * calculoResolucionArray[j]));
```

Por lo que cada punto es resultado de la suma de cada una de las amplitudes, multiplicada por el seno de la frecuencia asociada a ese voltaje.

Es importante destacar también el siguiente fragmento de código:

```
if (frecuenciasCopia[i] >= 10)
{
    while (frecuenciasCopia[i] >= 10)
    {
        frecuenciasCopia[i] /= 10;
    }
}
```

Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama

Este fragmento es para adaptar los datos a lo pedido por el DAC del generador. En concreto elimina los múltiplos en la frecuencia resultando una frecuencia de entrada en un rango de [0 a 10) Hertzios. Se descubrió que, si se generaba una onda sinusoidal de por ejemplo 10 KHZ, sin filtrar la frecuencia, posteriormente el DAC multiplicaba la frecuencia de origen por la generada, dando lugar a una onda sinusoidal 10000 KHZ.

Una pregunta que puede surgir al lector, es como no se hizo este filtrado de la frecuencia en el código de la clase correspondiente al osciloscopio Agilent. La respuesta es sencilla, se necesita su muestrear la onda, a una frecuencia menor para que el osciloscopio la genere correctamente, como el calculo de puntos se separó en otra clase, la única solución posible para que no afecte a la adaptabilidad del código es sobrecargar la función.

Finalmente citar, que para el simulador se creó una

Código de envío

El código para enviar una onda al DAC incluyendo la parte del filtrado es la siguiente:

```
const double MAXAMPLITUDSOPORTADA = 2.5;
double[] valores = new double[values.Length];
double frecuenciaC = maxfrecuencia;
while (frecuenciaC >= 10)
{
    frecuenciaC /= 10;
}
values.CopyTo(valores, 0);
if (maxamplitud <= MAXAMPLITUDSOPORTADA)
{
    for (int i = 0; i < valores.Length; i++)
    {
        valores[i] = Math.Round(valores[i], 2);
        valores[i] = valores[i] / maxamplitud;
    }
    try
    {
        // Esta secuencia, se tiene que realizar en este orden
        // Envio de datos
        // Activar interpolación
        // Indicar amplitud
        // Indicar frecuencia
        driver.WaveformGenerator.OutputEnabled = true;
        driver.WaveformGenerator2.Arbitrary.SetData(ref valores);
        driver.WaveformGenerator2.Arbitrary.Interpolation = true;
        driver.WaveformGenerator2.Amplitude = maxamplitud;
        driver.WaveformGenerator2.Frequency = maxfrecuencia/frecuenciaC;
    }
    ...
}
```

Tabla 59- Código correspondiente al envío de datos al osciloscopio

Inicialmente, se comprueba si el valor máximo de la amplitud del vector suministrado es mayor que la que soporta el dispositivo (2.5V) en cuyo caso se rechaza el envío y no se le envía al DAC.

Posteriormente, para facilitar el generado por parte del osciloscopio, los puntos suministrados se redondean a dos decimales, obteniendo no solo una respuesta más rápida sino una onda más “suave”. A continuación, una vez realizado el redondeo, se divide cada punto por el valor máximo de amplitud, para filtrar los valores y reducirlo al intervalo citado anteriormente de [0,1].

Una vez hecho esto, es preciso decirle al osciloscopio que reproduzca los puntos suministrados a una amplitud dada y a una frecuencia dada, lo que le permitirá al osciloscopio pasar del rango [0,1] a la onda original.

En cuanto a la frecuencia, como anteriormente los puntos se han generado a una frecuencia entre 0 y 10 Hz ahora es preciso reproducirla por el multiplicador. Así pues, si se ha creado una onda simple de 4kHz, la aplicación generará los puntos a 4Hz y el osciloscopio los reproducirá a 1 kHz obteniendo la onda generada.

En último lugar, citar que el orden necesario para el envío de datos es:

- 1. Activar la salida del generador de ondas**
- 2. Enviar los datos**
- 3. Activar la interpolación**
- 4. Indicar la amplitud máxima**
- 5. Indicar la frecuencia máxima**

6.3.1.2.Actualizar Datos

Para la actualización de los datos es preciso tener en cuenta que se ha de poder elegir entre una actualización continua de los mismos o una secuencial. Para ambos casos es preciso tener en cuenta el carácter multiparadigma de C# y por tanto la forma correcta de implementación de un esquema patrón observador utilizando delegados.

El código concreto que obtiene la distribución de puntos desde el osciloscopio, es el siguiente:

```
public Double[] actualizarDatos(String canal)
{
    bool ondaObtenida = false;
    string resultadosMedicion = "";

    System.Double[] WaveformArray = { 128, 64, 32, 16, 8, 4, 2, 1 };
    System.Double InitialX = 0;
    System.Double XIncrement = 0;
    ...
    driver.Acquisition.Type =
    AgInfiniiVisionAcquisitionTypeEnum.AgInfiniiVisionAcquisitionTypeAverage;
    driver.Acquisition.NumberOfAverages = 4;
    ...
    driver.Measurements.get_Item(canal).FetchWaveform(ref WaveformArray, ref InitialX, ref
    XIncrement);
    ondaObtenida = true;
    ...
    if (ondaObtenida == true)
    {
        WaveformArray[0] = Math.Round(WaveformArray[0], 3);
        // Se aplica el filtro gaussiano de sigma = 1 y mascara 1x3 a la onda
        for (int i = 2; i < WaveformArray.GetUpperBound(0) - 2; i++)
        {
            WaveformArray[i] = WaveformArray[i] * MascaraGauss[2];
            WaveformArray[i] += WaveformArray[i - 1] * MascaraGauss[1];
            WaveformArray[i] += WaveformArray[i - 2] * MascaraGauss[0];
            WaveformArray[i] += WaveformArray[i + 1] * MascaraGauss[3];
            WaveformArray[i] += WaveformArray[i + 2] * MascaraGauss[4];
            WaveformArray[i] = Math.Round(WaveformArray[i], 3);
        }
    }
    ...
}
return WaveformArray;
}
```

Tabla 60 - Código correspondiente a la actualización de datos en las clases del modelo

La función “*actualizarDatos()*” recibe como parámetro el canal del que se desea recibir la onda y devuelve un vector con los datos del osciloscopio. Inicialmente, se indica al osciloscopio que se desea una salida promediada, para eliminar ruido, con un número de medias de 4. Después mediante la operación “*Measurements.get_Item(canal).FetchWaveform()*” se obtiene los valores deseados.

Una vez recibidos, se procede a un segundo filtrado donde se multiplica por una mascara de Gauss (para realizar un desenfoque gaussiano) y se redondea los valores recibidos a 3 decimales.

Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama

Estos valores son manejados desde la vista, en concreto la siguiente función:

```
private void actualizarGrafica()
{
    double escalaY=0;
    string valores = " ";
    if (bGuardarDatos.IsEnabled == false) { bGuardarDatos.IsEnabled = true; }
    datos = fachada.actualizarDatos((String)cCanales.SelectedItem);
    escalaY = fachada.getEscalaEjeY((String)cCanales.SelectedItem);
    iGraficas.dibujarGrafica(datos, escalaY, dibujarGrafica, lienzo);

    ...
}
```

Tabla 61- Código correspondiente a la actualización de datos en las clases de la vista

Una vez recibido los datos, y tal y como se especifica en el capítulo de diseño, se pasa el control de widget “canvas” a la capa de utilidades, mediante la sentencia:

```
iGraficas.dibujarGrafica(datos, escalaY, dibujarGrafica, lienzo);
```

Sin embargo, esto no garantiza un dibujo en tiempo real desligado de la capa del modelo, dado que el hilo principal de la vista sigue acoplado a la llamada de actualizar los datos.

Para solucionar este problema, se recurre a los delegados. Un delegado [58] es un tipo que representa referencias a métodos con una lista de parámetros determinada y un tipo de valor devuelto.

En el caso de la aplicación:

```
public delegate void ActualizarInterfaz();
ThreadStart comienzoHilo;
Thread hiloActualizarTiempoReal;
public VentanaPrincipal()
{
    ....
    comienzoHilo = new ThreadStart(comenzarHilo);
    hiloActualizarTiempoReal = new Thread(comienzoHilo);
    hiloActualizarTiempoReal.Start();
}

private void comenzarHilo()
{
    System.Timers.Timer temporizador = new System.Timers.Timer(1000);
    temporizador.Elapsed += new ElapsedEventHandler(_timer_Elapsed);
    temporizador.Enabled = true;
}

private void _timer_Elapsed(object sender, ElapsedEventArgs e)
{
    this.Dispatcher.BeginInvoke(new ActualizarInterfaz(this.actualizarGrafica), null);
}
```

Tabla 62- Código correspondiente a la actualización de datos usando un delegado

Se define el delegado, como un atributo y también un hilo y una función asociada a dicho hilo. En el constructor, se crea el hilo y se inicia. En su método asociado “comenzarHilo()” se crea un “timer” que invoca al delegado.

Nótese que el delegado aunque ya definido se crea y se asocia a la función de “*actualizarGrafica*” citada anteriormente, con esto se consigue separar del hilo principal, el muestreo de la información por parte del osciloscopio.

6.3.1.3.Reconstrucción de ondas

A la hora de extraer características no se pudo implementar por completo, dado que el alcance de este paso da para un TFM mayor. No obstante, en este apartado se aprovecha para explicar los aspectos implementados, en el capítulo séptimo se detallará posibles líneas para solucionar los problemas planteados.

De cara a extraer las características hay que resolver los siguientes problemas, en este orden:

- Filtrado de ruido.
- Reconstruir amplitudes de la onda recibida.
- Reconstruir frecuencias de la onda recibida.
- Calcular desfase

Filtrado de ruido

La señal recibida del osciloscopio posee una gran cantidad ruido, supone un inconveniente para calcular picos y frecuencias dando valores erróneos. Las ilustraciones han sido extraídas de [1].

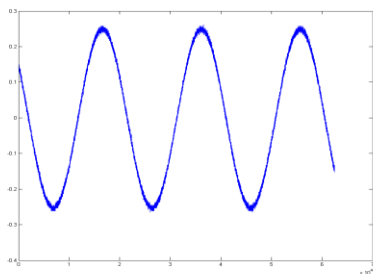


Ilustración 66- Captura de la señal sin zoom

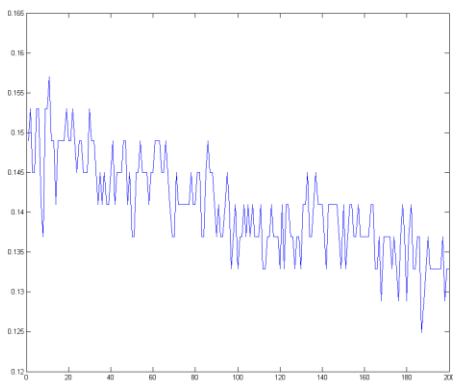


Ilustración 67-Ampliación de la señal anterior, hacia la mitad, con unas decenas de muestras

Por las figuras presentadas, se puede afirmar que el ruido posee un carácter discreto, es decir, la señal está escalonada y limitada únicamente a determinado rango de valores, esto lleva a deducir que el problema es del chip DAC del osciloscopio, concretamente en dos parámetros de diseño característicos de estos dispositivos como son:

- No-Linealidad Diferencial (DNL, Differential Non-Linearity).
- No-Linealidad Integral (INL, Integral Non-Linearity).

Se desarrolló inicialmente un algoritmo al que se denominó “algoritmo del faro”, este algoritmo funcionaba con señales de ruido escalonado, y consistía en recorrer el array de datos recibido realizando dos operaciones:

- Buscar el máximo
- Obtener la diferencia vertical, mínima y promedio entre un punto y el consecutivo.

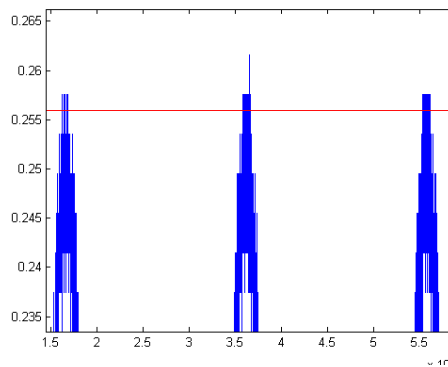


Ilustración 68-Demostración del algoritmo del “faro”, extraída de [1]

Una vez obtenido el punto más alto, se le restaba el valor de la diferencia promedio multiplicada por 1.5, obteniendo el nivel marcado por la línea roja de la figura. La finalidad de este resultado es obtener un límite inferior robusto para detectar picos. En una segunda iteración, se vuelve a recorrer el array de datos, cuando el voltaje supera o baja el nivel marcado por el umbral, se carga la posición horizontal de ese punto en un vector V. Si se revisa el vector V, se verá que existen diferentes conjuntos de valores muy próximos que corresponden a los valores que han quedado por encima de la línea, pero que, por efecto del ruido, suben y bajan. Entre estos conjuntos de valores existen saltos correspondientes a la diferencia entre picos, con ello ya se tendría sus posiciones horizontales y se podría calcular sus puntos medios para extraer el verdadero máximo.

Una segunda estrategia implementada, consistió en aplicar una matriz de convolución al vector devuelto, es decir una máscara aplicada a un vector lineal.

En concreto se efectuaron pruebas de aplicar un filtrado de medias y filtrado gaussiano, obteniendo mejores resultados con el gaussiano dado que es menos invasivo con la información original.

En último lugar se solicitó al osciloscopio que, por cada adquisición, tal y como se ha indicado antes, se realizara un promediado con un tamaño de $N=4$.

Reconstrucción de la amplitud

Se recuerda que la fórmula de una onda sinusoidal es la siguiente:

$$y(t) = A \sin(\omega t + \theta)$$

donde, y es el valor que toma la onda para un instante de tiempo t dado, A es la amplitud de la onda, ω la velocidad angular, y θ el desfase.

La amplitud de la onda sería la distancia entre el punto medio que alcanza la onda y el valor máximo que toma.

Cuando se analiza el vector de puntos de la onda, es decir, $y(t)$ es fácil recuperar la amplitud de la onda obteniendo el valor máximo que obtiene ese vector, menos el mínimo y dividirlo entre dos.

Así pues, el código utilizado para extraer la amplitud sería:

```
public void extraer(double[] puntos)
{
    double puntosMaximo= puntos.Max( );
    double puntosMinimo= puntos.Min( );
    Amplitud = ((puntosMaximo - puntosMin)/ 2) ;
}
```

Tabla 63- Código parcial de extraer características

Reconstrucción de la frecuencia

La reconstrucción de las frecuencias que componen la onda recibida es la parte más compleja y la que no se ha podido implementar.

Las razones, que llevan a esto son:

- La salida obtenida del osciloscopio es un vector de valores en punto flotante que representa voltaje, a ser unidimensional no es factible extraer el tiempo.
- El osciloscopio solo devuelve una ventana, definida por la escala del eje Y, y no la muestra completa, por lo que sumar el número de picos de la señal para obtener la frecuencia no es factible.
- Si el circuito altera la frecuencia original de la onda, algo deseable en un lock-in, no es posible a partir de las frecuencias de las ondas enviadas calcular las frecuencias de las ondas recibidas.

El desarrollo de soluciones que resuelvan este problema, exceden el tamaño de este proyecto, y se podrían plantear como trabajos futuros. En el capítulo séptimo de esta memoria se abordan posibles líneas de investigación para este apartado.

Reconstrucción del desfase

Aunque no ha sido posible implementar una solución para la reconstrucción de frecuencias, si ha sido posible hacerlo para el desfase, dado que matemáticamente no es necesaria la frecuencia para este cálculo.

Una vez calculada la amplitud A , y teniendo el vector con los datos recibidos del osciloscopio, se toma el valor en la posición 0 que se supone que es el obtenido para un instante de tiempo $t = 0$.

Teniendo en cuenta la ecuación de la onda: $y(t) = A \sin(\omega t + \theta)$ se calcula:

$$y(0) = A \sin(\omega 0 + \theta)$$

$$\begin{aligned}y(0) &= A \sin(\theta) \\ \frac{y(0)}{A} &= \sin(\theta) \\ \sin^{-1}\left(\frac{y(0)}{A}\right) &= \theta\end{aligned}$$

Y matemáticamente se habría obtenido el desfase.

Comprobando la documentación de las funciones matemáticas básicas proporcionadas en .NET Framework, se puede comprobar que las funciones “*Math.Asin()*” y “*Math.Sin()*” aceptan y devuelven los ángulos en radianes y no en grados como se hace en el sistema, es por ello necesario realizar unas conversiones antes y después de llamar a estas funciones usando las siguientes fórmulas:

$$\text{Radianes} = \text{Grados} * \frac{\pi}{180^\circ}$$

$$\text{Grados} = \text{Radianes} * \frac{180^\circ}{\pi}$$

El código resultante quedaría de la siguiente manera:

```
public void extraer(double[] puntos)
{
    Fase=Math.Asin(puntos[0]/Amplitud);
    Fase=Fase*(180/Math.PI);
    if (Fase<0)
    {
        Fase+=360;
    }
}
public double[] renderizar(double resolucion,double ciclos)
{
    double fase_radianes=Fase*(Math.PI/180);
    for(int j=straight_array.GetLowerBound(0);
j<=straight_array.GetUpperBound(0);j++)
    {
        sine_array[j]+=Amplitud*Math.Sin(((2f*Math.PI*frecuencia_local)*
straight_array[j])+fase_radianes);
    }
}
```

Tabla 64- Código parcial de extraer características y renderizar

Sin embargo, si se realiza algunas pruebas con el código, se aprecia que en cualquier ángulo entre 90° y 270° devuelve valores erróneos, es necesario por tanto calcular el cuadrante respecto al cual está el ángulo:

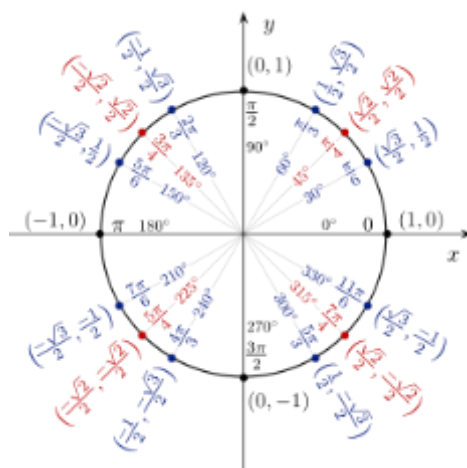


Ilustración 69- Diagrama de relación entre el número Pi y ángulos correspondientes, extraído de [1]

Para calcular el cuadrante correcto, es preciso saber si en los instantes próximos a $t = 0$, la onda está ascendiendo o descendiendo.

Las estrategias a seguir en este caso son:

- **Comprobar el valor inmediato a $t = 0$:**

Tiene la desventaja de que si la onda está afectada por el ruido, dará falsos valores.

- **Comprobar valores con un salto mayor:**

En caso de estar cerca de un pico, o en un valle el resultado sería erróneo.

- **Detectar el primer pico y el primer valle:**

Si, el instante $t=0$ está cerca del primer valle, implica que se está descendiendo, por el contrario, si está cerca de un pico estará ascendiendo.

Como desventaja citar que, la detección de valles no viene incorporada en las bibliotecas, aunque su implementación es fácil. Bastaría con invertir la onda, y recalculer el valor máximo o pico que toma.

Finalmente citar que para obtener la imagen especular de un vector respecto al eje Y, definido por $\frac{3\pi}{2}$ y $\frac{\pi}{2}$, se resta a cualquier ángulo el número π para obtener su mitad correspondiente que contiene al número 0.

Así pues, el código para calcular el desfase teniendo en cuenta el cuadrante sería:

```
public void extraer(double[]puntos)
{
    //Calculamos los picos utilizando el filtro de Savitzky-Golay
    DoubleVector dv= new DoubleVector(puntos);
    PeakFinderSavitzkyGolay peak_finder = new PeakFinderSavitzkyGolay(dv,5,4);
    peak_finder.LocatePeaks();
    double maxptos=puntos.Max();
    //"Invertimos" la onda inicial
    double[] reversed_ptos = new double[puntos.Length];
    for(int i=puntos.GetLowerBound(0); i<=puntos.GetUpperBound(0); i++)
        reversed_ptos[i]=maxptos-puntos[i];
    // Obtenemos los valles, sacando los picos de la onda invertida
    DoubleVector dv2= new DoubleVector(reversed_ptos);
    PeakFinderSavitzkyGolay valley_finder= new PeakFinderSavitzkyGolay(dv2,5,4);
    valley_finder.LocatePeaks();
    //Obtenemos la fase tal y como la devuelve el arcoseno
    Fase=Math.Asin(puntos[0]/Amplitud);
    //Si el primer pico esta mas cerca del primer dato que el primer valle
    //obtener la "imagenespecular" del angulo respecto del eje pi/2 a 3pi/2
    if(peak_finder[0].X > valley_finder[0].X)
        Fase=Math.PI-Fase;
    Fase=Fase*(180/Math.PI);
    if (Fase<0)
        Fase+=360;
}
```

Tabla 65- Código completo de extraer características

6.4. Cuarta Iteración

La cuarta iteración corresponde a mejoras realizadas en la tercera iteración y a otras funcionalidades estudiadas por mí y por mi compañero Eduardo en su respectivo TFM, que no se han implementado, o lo han sido parcialmente, pero que dado a que finalmente desbordaban los objetivos iniciales del proyecto, se plantean como líneas de futura investigación.

En concreto, en esta iteración están implementadas:

- Almacenar y recuperar ondas en formato XML.
- El proyecto ‘EsqueletoInicial’, que permite extraer características a partir de las ondas.

Y está planteado:

- Integrar el proyecto ‘EsqueletoInicial’ con el proyecto de la tercera iteración
- Implementar el recalcule de frecuencias expresado en la tercera iteración.
- Permitir, la integración de nuevos instrumentos al laboratorio.
- Calcular los barridos de frecuencias.

Almacenar y recuperar ondas en formato XML

Las necesidades del proyecto, llevaron a desarrollar un formato basado en XML para el almacenamiento y posterior recuperación de las ondas [1].

Las características de las ondas almacenadas en ficheros con este formato han de cumplir las siguientes condiciones:

- La extensión del fichero ha de ser, “.XML”.

Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama

- El fichero ha de estar codificado en UTF-8 (restricción impuesta por .NET Framework 4.5).
- La etiqueta “<wave_container>”, indica que ese fichero contendrá una serie de ondas simples.
- Las etiquetas “<wave>” y “</wave>” indican la apertura y cierre de una onda simple. Toda definición de onda simple empezará y terminará con esas etiquetas.
- La amplitud de una onda simple vendrá marcada entre las etiquetas “<amplitude>” y “</amplitude>”.
- La frecuencia de una onda simple vendrá marcada entre las etiquetas “<frequency>” y “</frequency>”.
- La fase de una onda simple vendrá marcada entre las etiquetas “<phase>” y “</phase>”.
- En caso de incluir magnitudes decimales, estas vendrán separadas por un “.”

Se incluye a continuación un ejemplo de una onda compuesta en formato XML con dos ondas simples:

```
<?xml version="1.0" encoding="utf-8"?>
<wave_container>
  <wave>
    <amplitude>2</amplitude>
    <frequency>6000</frequency>
    <phase>0</phase>
  </wave>

  <wave>
    <amplitude>3.5</amplitude>
    <frequency>7200</frequency>
    <phase>275</phase>
  </wave>
</wave_container>
```

Tabla 66- Ejemplo del esquema XML para almacenamiento de ondas

En concreto, no solo se ha implementado el almacenamiento sino también la recuperación de una onda a partir de un fichero XML, cuando se lee una onda en este formato automáticamente se crea un objeto de la clase “OndaCompuesta”, gracias además al patrón compuesto, esta clase creará por cada etiqueta “<wave>” contenida en el fichero, tantos objetos de la clase “Onda” asociados como sea necesario.

Las ventajas de este diseño son evidentes, permiten añadir funcionalidades (como el barrido), modificar de manera sencilla el experimento con ayuda de un editor, sin necesidad de recurrir a la aplicación principal...

Proyecto Esqueleto Inicial

En la tercera iteración, se comenzó a trabajar en paralelo en la clase “Onda”, en la extracción de características, y en los ficheros en XML. Se decidió, el crear un nuevo proyecto de nombre inicial “*EsqueletoInicial*” que incorporará estas características y incorporar la implementación una vez finalizada al proyecto general. De esta manera, las clases constituyentes podrían ser probadas y corregidas y no trasladar fallos así a la rama principal.

El algoritmo de “*EsqueletoInicial*” es el que se incluye a continuación:

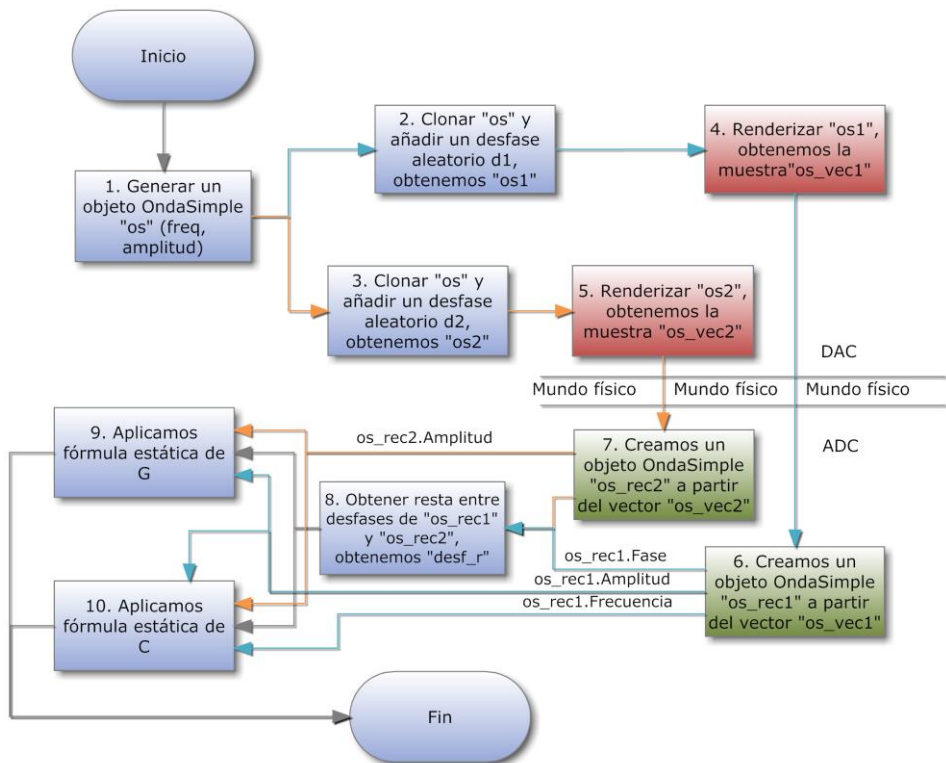


Ilustración 70 - Esquema de bloques del proyecto: "Esqueleto Inicial", extraído de [1]

Este algoritmo, funciona como procesamiento “*Offline*” de los datos, es decir, generando dos vectores de datos y a continuación procesándolos. Al obtener todos los datos es inmediato obtener el máximo real.

En el diagrama las líneas naranjas representan la salida del sistema, mientras que las azules representan las señales de referencia.

El algoritmo comienza creando una onda simple, de la clase “*Onda*”, a la que ajusta una frecuencia, una fase y una amplitud aleatoria, esta onda será la onda de referencia del lock-in virtual. A continuación, se clona el objeto anterior, y se modifica su fase de manera aleatoria, esta será la onda de salida del sistema que ha sido desfasada respecto a la de referencia. A posteriori ambas ondas son renderizadas, transformándose de objetos a vectores con valores de voltaje.

De estos dos vectores recibidos, se crean dos objetos “Onda” y a partir de la función “extraer” explicada anteriormente, se vuelve a obtener los valores de amplitud, y fase.
El siguiente paso, consiste en calcular la capacitancia de la salida del sistema como se especificó en el segundo capítulo de esta memoria.

El diagrama de clases para la gestión de ondas está reflejado en la siguiente figura:

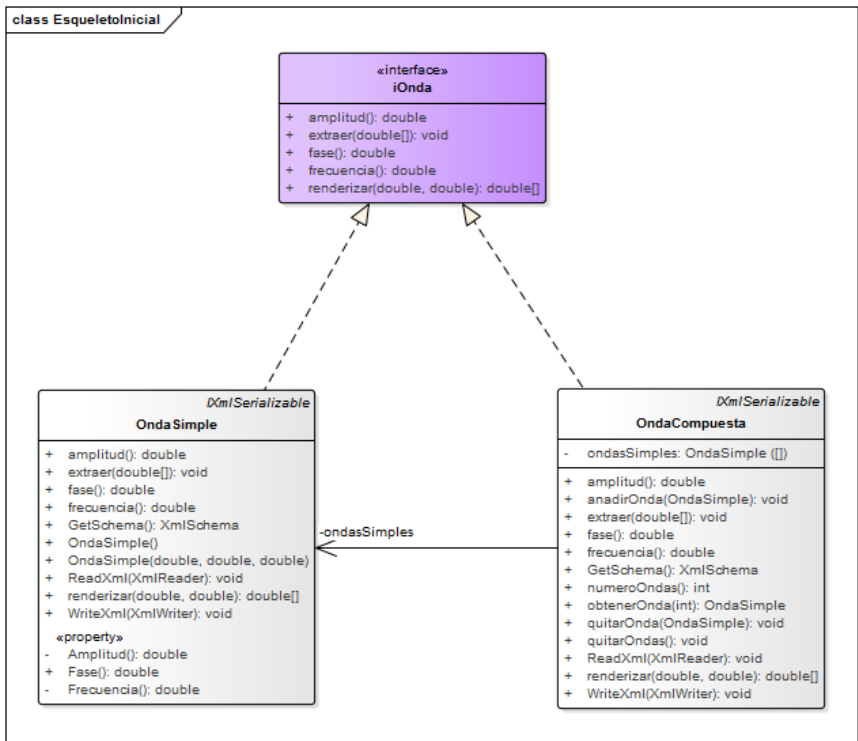


Ilustración 71- Esquema de clases correspondiente a la gestión de ondas, extraído de [1]

Existe una interfaz llamada “iOnda”, que define las características comunes a las ondas simples y compuestas. Todas las ondas, han de poder convertir un vector en definición de onda (“extraer”) y de transformar sus valores en un vector de valores de voltaje (“renderizar”).

Esta aplicación, “EsqueletoInicial”, utiliza los ficheros XML explicados anteriormente, creando un objeto del tipo “OndaCompuesta” que es capaz de extraer y crear las ondas simples que contenga. En caso de que solamente se desee una onda simple bastaría con definir una única etiqueta “<wave>” en el fichero, no obstante a través de la función “numerodeondas” es posible obtener el número de ondas simples de las que se compone la onda compuesta, por el contrario “obtenerOnda()” permitiría devolver la onda simple seleccionada.

La interfaz de este proyecto denominado “*EsqueletoInicial*” es la siguiente:

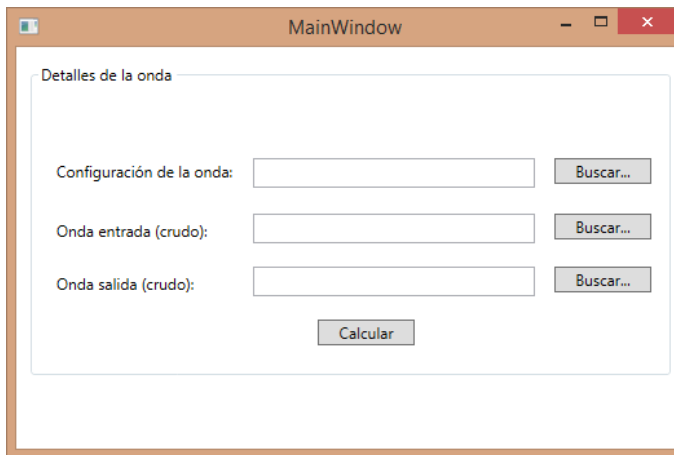


Ilustración 72- Aspecto de la interfaz del proyecto "EsqueletoInicial"

Como conclusión a este capítulo se incluye un diagrama de clases parcial [1], sobre las mejoras sin implementar, citadas anteriormente, sobre las que se incidirá más en el capítulo séptimo de esta memoria.

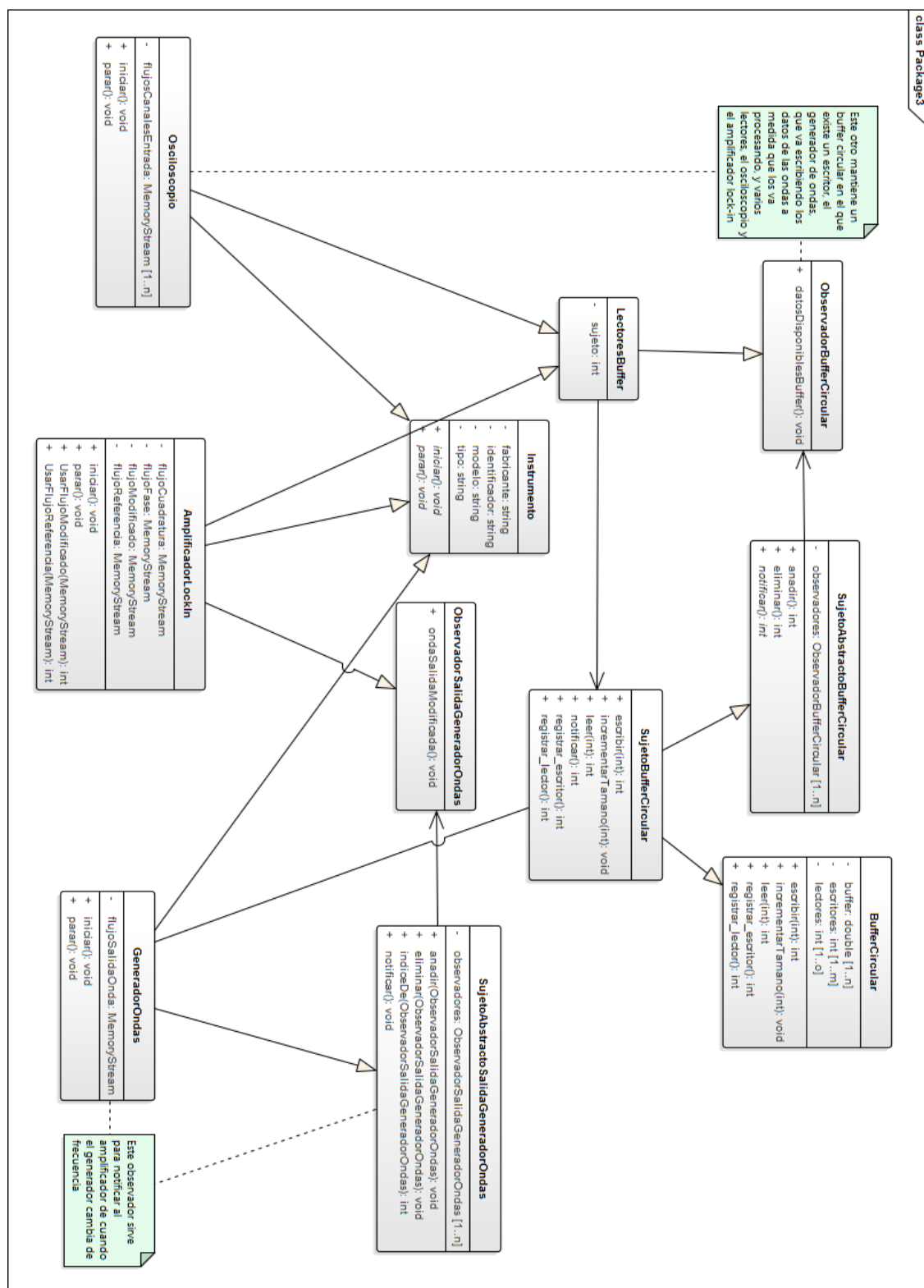


Ilustración 73- Diagrama de clases propuesto para la cuarta iteración

7. Capítulo séptimo: Pruebas y resultados

En este capítulo, se realizan pruebas de caja negra realizados a la aplicación, así como una serie de resultados obtenidos y su interpretación en el contexto de la electrónica.

Las pruebas de caja blanca (pruebas unitarias, de integración y sistema), que prueban los detalles procedimentales del software, y por tanto ligadas al código fuente, se han descartado debido a su complejidad y a que los resultados alargarían innecesariamente el contenido de esta memoria.

7.1. Pruebas de caja negra

Se organizará los casos de prueba en función de cada uno de los casos de uso.
La siguiente plantilla [47] se utilizará para definir cada prueba:

Prueba	Tipo	Resultado	Valoración
Descripción de la prueba a realizar.	Funcional(F)	Resultado esperado de la prueba.	Correcto(OK)
	Interfaz(I)		Incorrecto(NOK)

Tabla 67- Descripción de las pruebas a realizar

UC-001 – Crear Onda

Prueba	Tipo	Resultado	Valoración
Sin onda cargada, se introducen datos inválidos para la creación.	Interfaz(I)	La aplicación deniega la operación. Y no crea una onda nueva.	OK
Sin onda cargada, se introducen datos válidos para la creación.	Interfaz(I)	La aplicación acepta los datos introducidos, crea una nueva onda simple. A continuación actualiza la gráfica.	OK
Con una onda cargada, se selecciona añadir onda, habiendo introducido datos inválidos para la creación.	Interfaz(I)	La aplicación deniega la operación, y mantiene a la onda en su estado anterior.	OK

Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama

Con una onda cargada, se selecciona añadir onda, habiendo introducido datos válidos para la creación	Interfaz(I)	La aplicación acepta los datos introducidos, modifica la onda anterior añadiendo una nueva onda simple. A continuación actualiza la gráfica.	OK
--	-------------	--	----

Tabla 68- Prueba del caso de uso 001

UC-002 – Modificar Onda

Prueba	Tipo	Resultado	Valoración
Sin onda cargada se selecciona modificar onda.	Interfaz(I)	La aplicación deshabilita el botón asociado y deniega la operación..	OK
Con una onda cargada, se selecciona modificar onda, habiendo introducido datos inválidos para la modificación	Interfaz(I)	La aplicación deniega la operación, y mantiene a la onda simple seleccionada en su estado anterior.	OK
Con una onda cargada, se selecciona modificar onda, habiendo introducido datos válidos para la modificación	Interfaz(I)	La aplicación acepta la modificación, modifica la onda simple seleccionada. A continuación actualiza la gráfica.	OK

Tabla 69- Prueba del caso de uso 002

UC-003 – Borrar Onda

Prueba	Tipo	Resultado	Valoración
Sin onda cargada se selecciona borrar onda.	Interfaz(I)	La aplicación deshabilita el botón asociado y deniega la operación..	OK
Con una onda cargada, se selecciona borrar la última onda simple que la compone.	Interfaz(I)	La aplicación deshabilita el botón asociado a la operación y borra la onda. A continuación actualiza la gráfica.	OK
Con una onda cargada, se selecciona borrar una onda al azar de las simples que lo componen.	Interfaz(I)	La aplicación borra esa onda simple, a continuación actualiza la gráfica	OK

Tabla 70- Prueba del caso de uso 003

UC-004 – Cargar Onda

Prueba	Tipo	Resultado	Valoración
--------	------	-----------	------------

Se crea una onda en el mezclador de ondas, a continuación se cierra la aplicación y se vuelve a abrir..	Interfaz(I)	La aplicación almacena y recupera la onda creada antes de cerrar la aplicación.	OK
Sin onda cargada se carga un fichero con formato erróneo	Interfaz(I)	La aplicación deniega la operación y avisa al usuario de que el fichero es erróneo.	OK
Con una onda cargada se carga un fichero con formato erróneo	Interfaz(I)	La aplicación deniega la operación, avisa al usuario de que el fichero es erróneo, y a continuación vuelve a cargar la onda anteriormente establecida.	OK
Con una onda cargada se carga un fichero con datos válidos	Interfaz(I)	La aplicación lee el fichero correctamente, elimina la onda anteriormente establecida y calcula los nuevos valores de onda para la onda leída.	OK

Tabla 71- Prueba del caso de uso 004

UC-005 – Guardar Onda

Prueba	Tipo	Resultado	Valoración
Se crea una onda en el mezclador de ondas, a continuación, se cierra la aplicación.	Interfaz(I)	La aplicación almacena la onda creada antes de cerrar la aplicación.	OK
Sin onda cargada se selecciona guardar onda.	Interfaz(I)	La aplicación, deshabilita el botón asociado y deniega la operación.	OK
Con una onda cargada se selecciona guardar onda.	Interfaz(I)	La aplicación solicita la ruta del archivo a guardar. A continuación, guarda la onda creada.	OK

Con una onda cargada se eliminan todas sus ondas simples, a continuación, se selecciona guardar onda.	Interfaz(I)	La aplicación, deshabilita el botón asociado y deniega la operación.	OK
---	-------------	--	----

Tabla 72- Prueba del caso de uso 005

UC-006 – Extraer Características

Prueba	Tipo	Resultado	Valoración
Sin ficheros cargados de ondas de entrada, salida y configuración se selecciona calcular.	Interfaz(I)	La aplicación, deshabilita el botón asociado y deniega la operación.	OK
Con ficheros cargados de ondas de entrada, salida y configuración pero con datos inválidos, se selecciona calcular.	Funcional(F) Interfaz(I)	La aplicación detecta que los ficheros son erróneos, avisa al usuario y a continuación cancela la operación.	OK
Con ficheros cargados de ondas de entrada, salida y configuración y con datos válidos, se selecciona calcular.	Funcional(F) Interfaz(I)	La aplicación detecta efectúa los cálculos necesarios, y a continuación solicita al usuario la ruta del fichero donde exportar los datos..	OK

Tabla 73- Prueba del caso de uso 006

UC-007 – Conectar

Prueba	Tipo	Resultado	Valoración
Se selecciona el simulador como instrumento de conexión, a continuación se selecciona conectar.	Interfaz(I)	La aplicación se conecta con el simulador. A continuación deshabilita el botón de conectar	OK
Se selecciona el osciloscopio como instrumento de conexión, sin estar este físicamente conectado al puerto USB, a continuación	Funcional(F) Interfaz(I)	La aplicación detecta que el osciloscopio no está presente, informa al usuario y conecta con el simulador. A continuación	OK

Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama

se selecciona conectar		deshabilita el botón de conectar	
Se selecciona el osciloscopio como instrumento de conexión, estando este físicamente conectado al puerto USB, a continuación se selecciona conectar	Interfaz(I)	La aplicación conecta con el osciloscopio A continuación deshabilita el botón de conectar	OK
Estando conectada la aplicación al osciloscopio o al simulador, se selecciona conectar.	Interfaz(I)	La aplicación deshabilita el botón asociado y deniega la operación.	OK

Tabla 74- Prueba del caso de uso 007

UC-008 – Desconectar

Prueba	Tipo	Resultado	Valoración
Sin estar conectados al osciloscopio o al simulador, se selecciona desconectar.	Interfaz(I)	La aplicación deshabilita el botón asociado y deniega la operación.	OK
La aplicación está conectada al simulador y se selecciona desconectar.	Funcional(F) Interfaz(I)	La aplicación desconecta del simulador, a continuación deshabilita el botón de desconectar.	OK
La aplicación está conectada al osciloscopio y se selecciona desconectar.	Funcional(F) Interfaz(I)	La aplicación desconecta del osciloscopio, a continuación deshabilita el botón de desconectar.	OK

Tabla 75- Prueba del caso de uso 008

UC-009 – Actualizar Datos

Prueba	Tipo	Resultado	Valoración
Sin estar conectado al osciloscopio o al	Funcional(F)	La aplicación deshabilita el botón	OK

Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama

simulador, se selecciona “Actualizar Datos”.	Interfaz(I)	asociado, y deniega la operación..	
Con la aplicación conectada al osciloscopio, se selecciona “Actualizar Datos”.	Funcional(F) Interfaz(I)	La aplicación solicita los datos al osciloscopio para el canal seleccionado, a continuación muestra los valores recibidos y la gráfica asociada.	OK
Con la aplicación conectada al simulador, se selecciona “Actualizar Datos”.	Funcional(F) Interfaz(I)	La aplicación calcula los datos del simulador para el canal seleccionado, a continuación muestra los valores recibidos y la gráfica asociada	OK
Con la aplicación conectada al osciloscopio, se selecciona “Actualizar Datos en tiempo real”.	Funcional(F) Interfaz(I)	La aplicación, deshabilita el botón asociado a “Actualizar Datos”, a continuación, recibe de manera asíncrona pero permanente del osciloscopio los datos, a continuación, muestra los valores recibidos y la gráfica asociada.	OK
Con la aplicación conectada al osciloscopio, se desmarca la opción “Actualizar Datos en tiempo real”.	Funcional(F) Interfaz(I)	La aplicación, habilita el botón asociado a “Actualizar Datos”, a continuación, deja de recibir los datos de manera continuada.	OK

Tabla 76- Prueba del caso de uso 009

UC-0010 – Cambiar Canal

Prueba	Tipo	Resultado	Valoración
Con el simulador conectado, se cambia entre los cuatro canales posibles.	Funcional(F)	La aplicación obtiene y cambia entre los canales perfectamente.	OK

Con el osciloscopio conectado, se comprueba la lista de canales disponibles	Funcional(F)	La aplicación detecta y devuelve correctamente la lista de canales del osciloscopio.	OK
Con el osciloscopio conectado, se cambia a cada uno de los canales disponibles.	Funcional(F)	La aplicación cambia entre los canales perfectamente.	OK
Con el osciloscopio conectado, y la actualización en tiempo real marcada, se cambia a cada uno de los canales disponibles.	Funcional(F) Interfaz(I)	La aplicación cambia entre los canales parando la actualización durante el cambio y reanudándola para el canal seleccionado, perfectamente..	OK

Tabla 77- Prueba del caso de uso 010

7.2.Pruebas de generación y recepción de ondas

Para concluir este capítulo, se incluyen debido a la dificultad presentada, pruebas sobre la generación de ondas bajo el generador de DSO-X 3104A enviadas desde la aplicación y su comparativa con la misma señal generada automáticamente con el osciloscopio.

Montaje del experimento genérico

Para el experimento genérico de la aplicación, se ha realizado el siguiente montaje, que ilustra la fotografía a continuación:

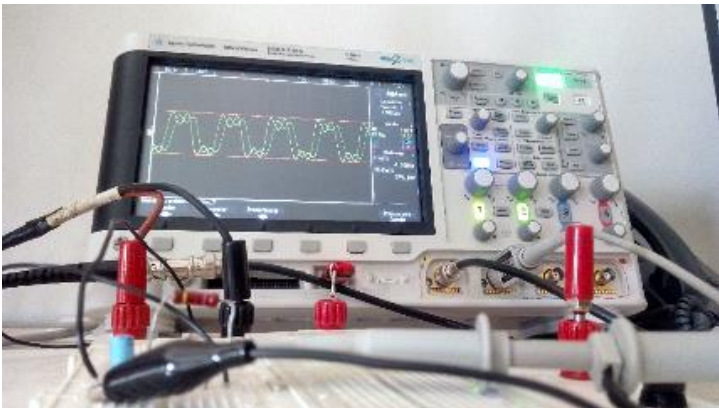


Ilustración 74- Fotografía que ilustra el experimento genérico del osciloscopio, cortesía de Francisco Javier Peña Rubio

Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama

El generador de ondas del osciloscopio, se ajusta al modo “*Arbitrary Waveform*” que recibe una serie de valores de voltaje en punto flotante, que se envían al driver, mediante las funciones citadas en el capítulo 6°. Los datos generados por el osciloscopio salen del conector BNC situado en la parte inferior izquierda, este conector es de tipo “T”, que replica la salida en dos direcciones, así en una de ellas irá directamente al canal 1 del osciloscopio para comprobar lo que se envía al mismo, y utilizarlo de canal de referencia. Por otro lado, la segunda salida es enviada al circuito RC, compuesto de dos resistencias y un condensador.

La señal una vez pasada del circuito varía en intensidad y pasa a un amplificador de transimpedancia (modelo “Stanford SR455A” [59] [14]) que amplifica el voltaje del mismo. La salida de este amplificador se conectaría al canal 2 del osciloscopio, y se recibiría por la aplicación para ser procesada como señal de salida del sistema.

Montaje para las pruebas de generación y recepción de ondas

Para la generación y recepción de ondas se ha realizado el siguiente montaje, expuesto en esta fotografía:



Ilustración 75- Fotografía que ilustra el experimento para la generación y recepción de ondas

En este caso, el montaje es más sencillo. Al igual que antes, el osciloscopio se configura en “*Arbitrary Waveform*”, de la salida del generador, se suprime el conector en forma de “T”, y se conecta un cable directamente al canal 1 del aparato.

Las pruebas a realizar, se componen de dos imágenes del osciloscopio, una en el modo “*Arbitrary Waveform*” con la onda enviada por la aplicación, y otra con la onda generada por el osciloscopio repitiendo las características enviadas.

Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama

Finalmente se recoge los resultados por cada prueba en una tabla que sigue la siguiente plantilla:

Prueba	Características electrónicas	Resultado	Valoración
Descripción de la prueba a realizar.	Amplitud(V)	Resultado esperado de la prueba.	Correcto(OK)
	Frecuencia(F)		Incorrecto(NOK)

Tabla 78- Descripción de la prueba de generación y recepción de ondas

Prueba 1

Prueba	Características electrónicas	Resultado	Valoración
Se envía una onda simple desde la aplicación, y se comprueba con la generada por el osciloscopio directamente.	La onda enviada tiene las siguientes características: 1.00(V) 10.00(kHZ)	Al ser la frecuencia, 1, no es necesario dividirla antes de enviarla, por lo que ambas señales coinciden.	Correcto(OK)

Tabla 79- Primera prueba

Imagen de la onda arbitraria



Ilustración 76 - Fotografía de la onda generada por la aplicación para la prueba 1

Imagen de la onda generada en modo seno

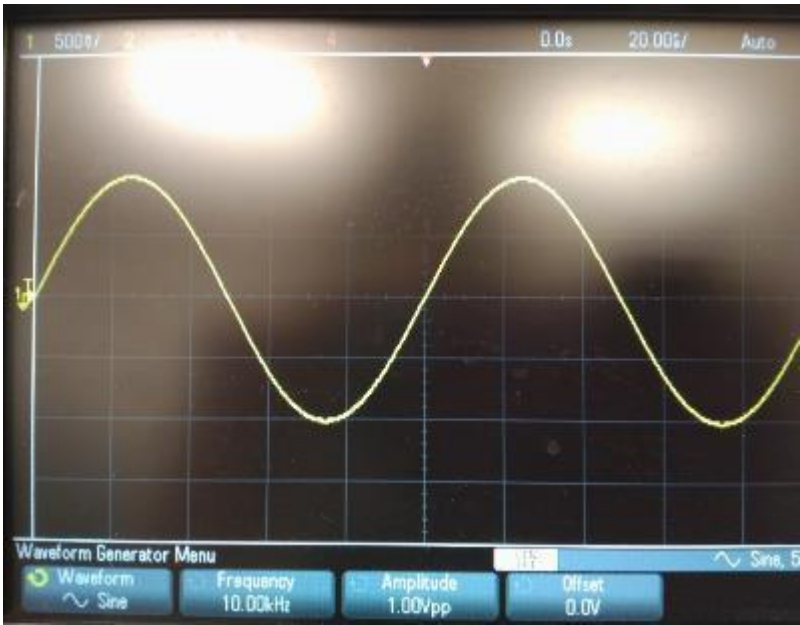


Ilustración 77-Fotografía de la onda generada por el osciloscopio para la prueba 1

Prueba 2

Prueba	Características electrónicas	Resultado	Valoración
Se envía una onda simple desde la aplicación, y se comprueba con la generada por el osciloscopio directamente.	La onda enviada tiene las siguientes características: 2.50(V) 50.00(kHZ)	Al ser la señal múltiplo de ‘5’, es necesario generar los puntos a esa frecuencia y luego en el envío reproducirlos a 10 kHz. Si se observan los medidores de ambas imágenes los valores de voltaje y frecuencia coinciden.	Correcto(OK)

Tabla 80- Segunda prueba

Imagen de la onda arbitraria



Ilustración 78-Fotografía de la onda generada por la aplicación para la prueba 2

Imagen de la onda generada en modo seno



Ilustración 79-Fotografía de la onda generada por el osciloscopio para la prueba 2

Prueba 3

Prueba	Características electrónicas	Resultado	Valoración
Se envía una onda simple desde la aplicación, y se comprueba con la generada por el osciloscopio directamente.	La onda enviada tiene las siguientes características: 0.55(V) 300.00(HZ)	Al ser la señal múltiplo de '3', es necesario generar los puntos a esa frecuencia y luego en el envío reproducirlos a 100 HZ. Si se observan los medidores de ambas imágenes los valores de voltaje y frecuencia coinciden.	Correcto(OK)

Tabla 81-Tercera prueba

Imagen de la onda arbitraria



Ilustración 80-Fotografía de la onda generada por la aplicación para la prueba 3

Imagen de la onda generada en modo seno

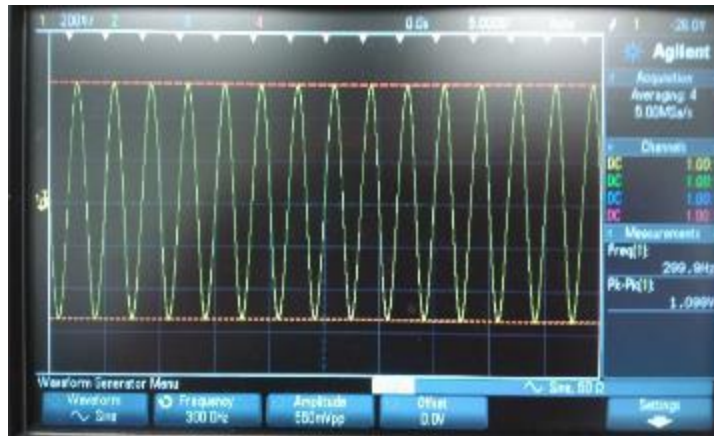


Ilustración 81- Fotografía de la onda generada por el osciloscopio para la prueba 3

Como conclusión citar, que las señales enviadas de carácter arbitrario, coinciden plenamente con las generadas por el osciloscopio en modo seno. Por lo que se puede afirmar que el algoritmo desarrollado de suma de senos funciona correctamente.

8. Capítulo Octavo: Conclusiones

En este capítulo, se detallan las dificultades encontradas, los objetivos alcanzados, las líneas de futuro trabajo y las conclusiones extraídas.

8.1. Dificultades encontradas

- **Falta de documentación:**

La mayor dificultad encontrada ha sido la falta de documentación, proporcionada por el fabricante para su desarrollo. Inicialmente se sufrió a la hora utilizar los drivers IVI, para tratar de controlar el osciloscopio Agilent, en concreto cuando se decidió realizar el proyecto con Matlab, y la única referencia encontrada fue el fragmento de código proporcionado que se utilizó como base, como ya se ha citado anteriormente en la memoria, la API en Matlab estaba parcialmente implementada lo que llevó a derivar el proyecto a C#.

- **Falta de tiempo de dedicación al proyecto:** Debido al contrato que tenía de trabajo durante el año pasado a jornada completa y al Máster GAIA que estaba cursando por la Universidad Alcalá de Henares, fue complicado compaginar el proyecto con estas otras tareas, tal y como aparece reflejado en el capítulo tercero de esta memoria. La consecuencia inmediata, es que supuse un retraso en el TFM relacionado de mi compañero Eduardo Rodríguez Gutiez. Una solución aplicada a esto, fue instalar en el ordenador de mesa un gestor de repositorios de Microsoft, llamado “*Team Foundation Server*”, para hacer más práctico la manipulación del código por ambos miembros del equipo. Adicionalmente se instaló un servidor GIT en el mismo ordenador, ya no solo para compartir el código sin necesidad de Visual Studio, sino que también para compartir documentación y bibliografía
- **Organización del proyecto:** Pese al sistema de control de versiones era necesario añadir un esquema organizativo, que permitiera conocer en qué tareas estábamos trabajando y en cuales estaba cada miembro del equipo desarrollando. Se recurrió finalmente a aplicar KanBan, para lo cual se creó un tablero en la corchera del laboratorio donde se dejaron los distintos esquemas como pueden ser diagramas de flujo y clases, y también algoritmos en pseudocódigo.
- **Ingeniería inversa sobre el envío de ondas:** Cuando la aplicación enviaba las ondas al generador, esta no las procesaba correctamente, debido a que cómo se citó en anteriores capítulos, el DAC generaba los puntos de voltaje en un rango determinado [0..1] sobre el cual había además que computar la frecuencia generando los puntos a la frecuencia final y submuestreándoles en el envío. Estas conversiones, debido a la falta de documentación, no fueron inmediatas y supusieron un retraso en el proyecto. Se llegó a enviar un correo a la sede de Keysight en España para preguntarles sobre como realizar esta conversión, no habiendo obtenido respuesta aún.
- **Falta de recursos inicial:** Al principio del proyecto, se disponía en el laboratorio de un ordenador con procesador Pentium IV que apenas tenía potencia suficiente como para

ejecutar la última instancia de Matlab, y se congelaba al abrir una segunda aplicación. Los tutores de este proyecto, solventaron el problema adquiriendo un ordenador con procesador i5 que está listado en los costes.

- **Reconstrucción de onda:** Como se ha expuesto anteriormente en anteriores capítulos, la señal recibida a través del osciloscopio, no nos permitía caracterizar la onda, por lo que hubo que desarrollar algoritmos, para a partir de los puntos recibidos extraer las amplitudes y frecuencias de las que la onda se compone. Esto se logró solo para la amplitud, dado que para la frecuencia se requiere un estudio que llevaría a la creación de otro trabajo final de máster.

En general, este proyecto ha tenido una dificultad elevada, debido en parte en que no se tenían conocimientos de procesamiento digital de señales. Gracias a la colaboración de nuestros tutores, y de otros profesores de la Escuela, como son Héctor García, Jesús Arias y Manuel González. Ha sido posible desarrollar el proyecto hasta su grado de desarrollo a escritura de estas líneas.

8.2. Objetivos alcanzados

Los objetivos alcanzados durante el desarrollo de este proyecto han sido:

- Se ha diseñado una aplicación orientada a objetos extensible, mediante el uso de una API y el patrón fachada, se pueden añadir múltiples tipos de dispositivos de los cuales ya forman parte el generador de ondas, el osciloscopio y el simulador. Además, la aplicación se ha codificado en C#, que es una variante de C, lo que permite su adaptación y portabilidad a otros sistemas inclusive dispositivos embebidos.
- Se ha conseguido implementar un generador de señal que permite crear, modificar y guardar frecuencias simples, complejas y barridos.
- Se ha conseguido implementar un analizador de señal que permite extraer las amplitudes de una señal a partir de un conjunto discreto de datos.
- Se ha logrado extraer, a partir de dos señales sinusoidales (referencia y salida del sistema), sus componentes en fase y cuadratura.
- Se ha creado y desarrollado diferentes estrategias para reducir el coste computacional de los algoritmos utilizados.

8.3. Líneas de trabajo futuro

En el siguiente apartado se exponen las posibles líneas de trabajo futuro de este proyecto:

- La primera línea de trabajo consiste en desarrollar algoritmos que permitan la reconstrucción de frecuencias, a partir el conjunto discreto de puntos recibidos por el osciloscopio. Hay dos posibles aproximaciones a este trabajo:
 - Desarrollar el algoritmo utilizando únicamente la ventana de datos recibida, y introduciendo el modelo matemático del circuito sobre el cual se aplica la señal de entrada y comparar con esta.
 - Utilizar los medidores proporcionados por el osciloscopio Agilent, y ir restando a la onda compuesta sucesivas ondas simples con la frecuencia proporcionada por el medidor, hasta obtener la lista completa.
- La segunda línea de trabajo tendría que integrar los dos proyectos paralelos presentados en la tercera iteración, en uno solo. Por otro lado, convendría añadir lo especificado en la línea primera de trabajo para tener un producto integro que cumpla con todas las especificaciones de desarrollo. También convendría el desarrollo de pruebas unitarias para verificar las funciones y clases de esta nueva aplicación.
- Una tercera línea de trabajo, aprovechando la adaptabilidad del código, sería convertir la aplicación en una suite de trabajo siguiendo los “mock-ups” presentados en el capítulo de diseño. Es conveniente citar que es necesario el desarrollo de drivers para controlar estos dispositivos.
- La cuarta línea de trabajo tiene que ver con la portabilidad del código, en concreto convendría en caso de uso en dispositivos embebidos como puede ser un procesador digital de señal, conviene portar el código de la extracción de características a C++ o C, implementando no solo el código de la aplicación sino el de las bibliotecas externas.
- En el proyecto de la tercera iteración “*EsqueletoInicial*” se ha utilizado la biblioteca “*NMath*” para el cálculo de los picos mediante el detector de Savitky-Golay. Estas librerías están optimizadas para procesadores Intel y para arquitecturas paralelizables, por lo que sería de interés de estudio como poder paralelizar el computo en caso de continuar con el desarrollo de este proyecto.
- En último caso, se podría desarrollar utilizando un mini-computador tipo Intel Edison o Raspberry Pi, un dispositivo hardware que permitiera la extracción de características a partir de la lectura de un osciloscopio, a este dispositivo se le podría añadir una pantalla LCD o similar para controlarlo, creando así un instrumento que combinara la funcionalidad de los dispositivos que forman parte de la aplicación.

8.4. Conclusiones

Aunque a priori, la temática de este trabajo fin de máster pueda parecer que se aplica a conjunto cerrado de aplicaciones del área de la electrónica, no es verdad, las señales y su procesamiento tienen su reflejo inmediato en áreas de la informática como son, el audio digital o el procesamiento de imagen y las aplicaciones que hay detrás de ellos.

A mayores, hemos podido ampliar los conocimientos, tanto mi compañero como yo, sobre gestión de proyectos, documentación y sobre todo electrónica y programación.

Es la primera vez, que hemos trabajado con un amplificador Lock-in, entendiendo su funcionamiento y necesidad de desarrollar un modelo software que reduzca el coste de adquirir un dispositivo de estas características en un laboratorio de electrónica. Por otro lado, el Lock-in tiene aplicación no solo en electrónica, sino también en física, en química y en telecomunicaciones.

De la misma manera, nunca antes habíamos trabajado con tanta profundidad con un instrumento como es el osciloscopio, anteriormente en la ingeniería técnica, habíamos utilizado este dispositivo en las asignaturas de Sistemas Digitales, Periféricos, Electrónica y Física. Pero siempre a un nivel más básico que el alcanzado durante el desarrollo de este proyecto. También hemos tenido la oportunidad de repasar conocimientos de electrónica de corriente alterna, como son la impedancia, el ‘ancho de banda’, o el análisis de Fourier.

También es la primera ocasión en la que he trabajado con herramientas colaborativas, como “*Team Foundation Server*” y “*Bonobo Git Server*”, que han servido para poder trabajar con el código donde cada “*commit*” o subida ha permitido olvidarnos de incidencias de poder sobrescribir el código del compañero. También “*Bonobo*” ha permitido tener un espacio de compartir documentos de cara a la documentación de este proyecto.

Por último, citar que, aunque no se han podido resolver todos los problemas planteados, en concreto falta de reconstrucción de frecuencias, si se han podido implementar soluciones para el resto ofreciendo una solución global al problema planteado. No obstante, recordar, que la reconstrucción de frecuencias ha sido un problema con el que nos hemos encontrado durante el desarrollo y que constituye un problema matemático sobre como reconstruir datos continuos a partir de un conjunto discreto, más que sobre una cuestión acerca de electrónica.

Contenido del CD-ROM

El CD-ROM adjunto contiene los siguientes archivos:

- Memoria del Trabajo Fin de Máster “Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama en formato PDF. Se encuentra en la carpeta:” *Memoria*”
- Presentación en PowerPoint, del Trabajo Fin de Máster “Generación de señales arbitrarias y medida de corriente con osciloscopio digital de alta gama”. Se encuentra en la carpeta: *Presentación*”
- Diagramas de UML en formato PNG. Se encuentran en la carpeta: *Diagrama*”
- Código fuente del proyecto “Análisis Espectral de la Impedancia”. Se encuentra en la carpeta: *Proyecto\ProyectoRefactorizado*”
- Código fuente del proyecto “EsqueletoInicial”. Se encuentra en la carpeta: *Proyecto\EskeletoInicial*”
- Código fuente correspondiente a la primera iteración. Se encuentra en la carpeta: *Proyecto\PrimeraIteración*”.
- Instalable del proyecto “Análisis Espectral de la Impedancia”. Se encuentra en la carpeta: *Instalables\ProyectoRefactorizado*”.
- Instalable del proyecto “EsqueletoInicial”. Se encuentra en la carpeta: *Instalables\EskeletoInicial*”.
- Librerías de Keysight y Agilent necesarias para ejecutar el proyecto. Se encuentran en la carpeta: *Instalables\Librerias*”

Los requisitos mínimos para poder ejecutar la aplicación son:

- Procesador a 1 gigahercio (GHz) o más rápido de 32 bits (x86) o de 64 bits (x64)*
- 1 gigabyte (GB) de RAM (32 bits) o 2 GB de RAM (64 bits)
- 100 MB de espacio disponible en el disco duro.

La versión mínima de Visual Studio necesaria para abrir los archivos del proyecto es: Visual Studio 2010.

Bibliografía

- [1] E. Rodríguez Gutiez, H. Castán Lanaspá y S. Dueñas Carazo, Implementación software de un amplificador Lock-in, Valladolid: Univesidad de Valladolid, 2015.
- [2] Keysight Technologies, «Oscilloscope | Keysight (Agilent),» Keysight Technologies 2000-2016, 2016. [En línea]. Available: <http://www.keysight.com/en/pcx-x2015004/oscilloscopes?cc=ES&lc=spa>. [Último acceso: 18 02 2016].
- [3] I. Lewis, «Adapter Pattern,» Wikipedia, 26 07 2002. [En línea]. Available: https://en.wikipedia.org/wiki/Adapter_pattern. [Último acceso: 18 02 2016].
- [4] Grupo de Caracterización de Materiales y Dispositivos Electrónicos, Departamento de Electrónica, Universidad de Valladolid, «Grupo de Caracterización de Materiales y Dispositivos Electrónicos,» Universidad de Valladolid, [En línea]. Available: <http://lab11.ele.uva.es/>. [Último acceso: 17 02 2016].
- [5] National Instruments, «Software de Desarrollo de Sistemas NI LabVIEW,» 2016. [En línea]. Available: <http://www.ni.com/labview/esa/>. [Último acceso: 17 02 2016].
- [6] Keysight Technologies, «2000, 3000, 4000, 6000 InfiniiVision X-Series Oscilloscope IVI and MATLAB Instrument,» Keysight Technologies, 29 10 2015. [En línea]. Available: <http://www.keysight.com/main/software.jsp?ckey=2019021&lc=spa&cc=ES&nid=-33573.1006712&id=2019021>. [Último acceso: 30 10 2015].
- [7] Agilent Technologies, Agilent InfiniiVision 3000 X-Series Oscilloscopes User's Guide, Malaysia: Agilent Technologies, 2013.
- [8] Stanford Research Systems, Model SR844 RF Lock-In Amplifier, 2.8 ed., Sunnyvale, California: Stanford Research Systems, 2013.
- [9] T. Ruiz Vázquez, O. Arbelaitz Gallego, I. Etxebarria Uztarroz y A. Ibarra Lasa, Análisis Básico de Circuitos Eléctricos y Electrónicos, Madrid: Pearson Educación, 2004.
- [10] A. Barahona, C. Herrera y Y. Villota, Señales Análogas, 2015.
- [11] A. Fernández González, «Superposición de ondas,» Universidad de Sevilla, 18 03 2010. [En línea]. Available: http://laplace.us.es/wiki/index.php/Superposici%C3%B3n_de_ondas. [Último acceso: 30 01 2016].
- [12] Wikipedia, «Osciloscopio,» Wikipedia, 9 07 2003. [En línea]. Available: <https://es.wikipedia.org/wiki/Osciloscopio>. [Último acceso: 20 01 2016].
- [13] Keysight Technologies, Evaluating Oscilloscope Fundamentals Application Note, Keysight Technologies, 2014.
- [14] Agilent Technologies, Osciloscopios Agilent InfiniiVision 3000 serie X, Séptima ed., Malasia: Agilent Technologies, Inc., 2013.
- [15] Keysight Technologies, Keysight InfiniiVision 3000 X-Series Oscilloscopes Programmer's Guide, Colorado Springs: Keysight Technologies, 2015.
- [16] ThinkSRS, «About Lock-In Amplifiers Application Note #3,» 2010. [En línea]. Available: <http://www.thinksrs.com/downloads/PDFs/ApplicationNotes/AboutLIAs.pdf>. [Último acceso: 22 02 2016].
- [17] University of Toronto, «Lock-In Amplifier Quick Start,» Youtube, Toronto, 2015.
- [18] R. Scholten, «Lock-In Amplifiers,» 02 2011. [En línea]. Available:

- http://electronics.physics.helsinki.fi/wp-content/uploads/2011/02/lockin_rob_web.pdf. [Último acceso: 14 02 2016].
- [19] S. Mingxin y L. Fangfang, Design and Implementation of the Phase-locked Amplifier, Harbin: Harbin University of Science and Technology, 2015.
- [20] Stanford Research Systems, MODEL SR850 - DSP Lock-In Amplifier, 2.0 ed., SunnyVale, California: Stanford Research Systems, 2009.
- [21] Wikipedia, «RC Time Constant,» 11 02 2015. [En línea]. Available: https://en.wikipedia.org/w/index.php?title=RC_time_constant&oldid=646605329. [Último acceso: 23 02 2016].
- [22] Wikipedia, «Filtro Paso Bajo,» 15 11 201. [En línea]. Available: https://es.wikipedia.org/w/index.php?title=Filtro_paso_bajo&oldid=78160694. [Último acceso: 23 02 2016].
- [23] Á. Domenech, «Efectos de sonido: la ecualización,» Universitat Politècnica de Valencia, [En línea]. Available: <http://www.disca.upv.es/adomenec/IASPA/tema4/ES-Filtros.html>. [Último acceso: 23 02 2016].
- [24] W. Storr, «Low Pass Filter - Passive RC Filter Tutorial,» [En línea]. Available: http://www.electronics-tutorials.ws/filter/filter_2.html. [Último acceso: 21 02 2016].
- [25] L. Jure López, «Clase 10: Introducción a los filtros digitales,» 04 12 2011. [En línea]. Available: <http://www.eumus.edu.uy/eme/ensenanza/electivas/dsp/presentaciones/clase10.pdf>. [Último acceso: 15 02 2016].
- [26] M. Aguilar Gutiérrez, Bioelectromagnetismo: campos eléctricos y magnéticos y seres vivos, 1ª ed., Madrid: Editorial CSIC - CSIC Press, 2001.
- [27] Wikipedia, «Ruido Rosa,» 10 01 2014. [En línea]. Available: https://es.wikipedia.org/w/index.php?title=Ruido_rosa&oldid=71832206.
- [28] W. Tomasi, Sistemas de comunicaciones electrónicas, 4ª ed., New Jersey: Pearson Education, 2003.
- [29] J. Beasley, J. D. Hymer y G. M. Miller, Electronic Communications: A System Approach, 1ª Edición ed., Prentice Hall.
- [30] A. De Marcellis, A. Di Giansante, G. Ferri, C. Di Natale, E. Martinelli y A. D' Amico, «Analog automatic lock-in amplifier for very low gas concentration detection,» *ScienceDirect*, p. 5, 2010.
- [31] F. Hoehne, L. Dreher, J. Behrends, M. Fehr, H. Huebl, K. Lips, A. Schnegg, M. Suckert, M. Stutzmann y M. S. Brandt, «Lock-in detection for pulsed electrically detected magnetic resonance,» *AIP Review of Scientific Instruments*, 20 04 2012.
- [32] G. Przemyslaw, «EPR spectrometer,» Wikimedia Commons, Kraków, 2003.
- [33] Physics 111-Lab Wiki, «Nuclear Magnetic Resonance - Physics 111-Lab Wiki,» Physics 111-Lab Wiki, 1 04 2015. [En línea]. Available: http://www.advancedlab.org/mediawiki/index.php/Nuclear_Magnetic_Resonance. [Último acceso: 23 02 2016].
- [34] A. Abu_Al_Aish, M. Rehman, M. Z. Abdullah y A. H. Abu Hassan, «Microcontroller Based Capacitive Mass Measuring System,» *MEASUREMENT SCIENCE REVIEW*, vol. 10, nº 1, pp. 15-18, 2 02 2010.
- [35] L. Shaoyang y W. Yifen, «Chapter 6 – A Review of the Application of Atomic Force Microscopy (AFM) in Food Science and Technology,» *Elsevier*, vol. 62, pp. 200-241, 2011.
- [36] A. M. Gigler, C. Dietz, M. Baumann, N. F. Martinez, R. García y R. W. Stark, «Repulsive bimodal atomic force microscopy on polymers,» *Beilstein J. Nanotechnol*, vol. 3, pp. 456-463, 2012.

- [37] H. Scott, «Making sound waves visible with a lock-in amplifier, microphone, and LEDs,» New York, 2012.
- [38] I. Sommerville, Ingeniería del Software, 7ª Edición ed., P. A. Wesley, Ed., Pearson Education, 2005.
- [39] J. M. Marques Corral, Apuntes de Ingeniería del Software I, Valladolid: Universidad de Valladolid, 2006.
- [40] «SDLC - Waterfall Model,» Tutorials Point, 2014. [En línea]. Available: http://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm. [Último acceso: 17 12 2015].
- [41] J. Arlow y I. Neustadt, Programación en UML 2, Madrid: Anaya, 2006.
- [42] H. Hristo Toshkov, «Introduction to OpenUp,» Eclipse Foundation, 30 05 2012. [En línea]. Available: <http://epf.eclipse.org/wikis/openup/>. [Último acceso: 18 12 2015].
- [43] K. Schwaber y J. Sutherland, «The Scrum Guide Online,» Scrum.org, 2014. [En línea]. Available: <http://www.scrumguides.org/scrum-guide.html>. [Último acceso: 18 12 2015].
- [44] H. Kniberg y M. Skarin, Kanban y Scrum - Obteniendo lo mejor de ambos, Estados Unidos de América: C4Media, 2010.
- [45] Keysight Technologies, «82357B USB/GPIB Interface High-Speed USB 2.0,» Keysight Technologies, 2015. [En línea]. Available: <http://www.keysight.com/en/pd-851808-pn-82357B/usb-gpib-interface-high-speed-usb-20?cc=US>. [Último acceso: 22 03 2015].
- [46] Fundación Universidad Empresa, «Programa GAIA,» 12 2014. [En línea]. Available: <http://programagaia.es/>. [Último acceso: 10 12 2014].
- [47] A. Escobar Cotán, M. Á. Laguna Serrano y J. Finat Codes, Monitorización de ejercicios físicos y detección de caídas usando Kinect: Moca Kinect, Valladolid, Valladolid: Universidad de Valladolid, 2013.
- [48] MSDN Developer Network, «Introducción (WPF),» Microsoft Corporation, 2010. [En línea]. Available: [https://msdn.microsoft.com/es-es/library/ms742119\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/ms742119(v=vs.110).aspx). [Último acceso: 20 02 2016].
- [49] MSDN Developer Network, «The Model-View-Presenter (MVP) Pattern,» Microsoft Corporation, 2008. [En línea]. Available: <https://msdn.microsoft.com/en-us/library/ff649571.aspx>. [Último acceso: 20 02 2016].
- [50] T. Snyder, «MVC or MVP Pattern – Whats the difference?,» Infragistics, 10 08 2010. [En línea]. Available: http://www.infragistics.com/community/blogs/todd_snyder/archive/2007/10/17/mvc-or-mvp-pattern-whats-the-difference.aspx. [Último acceso: 20 02 2016].
- [51] A. Martínez Monés, «Apuntes de Ingeniería de la Interacción,» Universidad de Valladolid, Valladolid, 2013.
- [52] A. Pérez Garro, «Modern UI,» Wikipedia, 19 06 2014. [En línea]. Available: https://es.wikipedia.org/wiki/Modern_UI. [Último acceso: 20 02 2016].
- [53] MathWorks, «MATLAB- El Lenguaje de cálculo técnico,» The MathWorks, Inc., 2016. [En línea]. Available: http://es.mathworks.com/products/matlab/index.html?s_tid=gn_loc_drop. [Último acceso: 20 02 2016].
- [54] IVI Foundation, «Shared Components,» IVI Foundation, [En línea]. Available: http://www.ivifoundation.org/shared_components/. [Último acceso: 25 02 2016].
- [55] Keysight Technologies, «IO Libraries Suite,» Keysight Technologies, [En línea]. Available: <http://www.keysight.com/en/pd-1985909/io-libraries-suite?cc=ES&lc=eng>. [Último acceso: 10 02 2016].
- [56] Keysight Technologies, «33503A BenchLink Waveform Builder Pro Software,» Keysight Technologies, 2010. [En línea]. Available: <http://www.keysight.com/en/pd-1962285-pn->

33503A/benchlink-waveform-builder-pro-software?cc=ES&lc=eng. [Último acceso: 10 02 2016].

- [57] C. Vinod, «Agilent Infiniium and InfiniiVision Oscilloscopes – MATLAB Example,» The MathWorks, Inc., [En línea]. Available: <http://www.mathworks.com/matlabcentral/fileexchange/22485-agilent-infiniium-and-infiniivision-oscilloscopes---matlab-example>. [Último acceso: 20 10 2014].
- [58] MSDN Developer Network, «Tutorial de delegados,» Microsoft Corporation, [En línea]. Available: [https://msdn.microsoft.com/es-es/library/aa288459\(v=vs.71\).aspx](https://msdn.microsoft.com/es-es/library/aa288459(v=vs.71).aspx). [Último acceso: 1 03 2016].
- [59] Stanford Research Systems, MODEL SR445A 350 MHz PREAMPLIFIER, 1.1 ed., Sunnyvale, California: Stanford Research Systems, 2006.